Sylvie Gibet
Nicolas Courty
Jean-François Kamp (Eds.)

Gesture in Human-Computer Interaction and Simulation

6th International Gesture Workshop, GW 2005
Berder Island, France, May 2005
Revised Selected Papers

Springer

Lecture Notes in Artificial Intelligence 3881

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Sylvie Gibet   Nicolas Courty
Jean-François Kamp (Eds.)

Gesture in Human-Computer Interaction and Simulation

6th International Gesture Workshop, GW 2005
Berder Island, France, May 18-20, 2005
Revised Selected Papers

Preface

The international Gesture Workshops have become the leading interdisciplinary events for dissemination of the latest results on gesture-based communication. The goal of these workshops is to bring together researchers who want to meet and share ideas on advanced research on gesture related to multidisciplinary scientific fields. Depending on the fields, the objectives can be very different. While physiology and biomechanics aim to extract fundamental knowledge of physical gesture, computer sciences try to capture different aspects of gesture and extract features that help to identify, interpret or rebuild the underlying mechanisms of communication gestures. Other approaches and methodologies are followed by cognitive sciences and linguistics, bringing a complementary understanding of motor control and gesture meaning. The results can be enhanced by technological applications or demonstrations. For example, gestural interaction in an augmented or virtual reality context leads to active application areas. Since 1996 gesture workshops have been held approximately every two years, with full post-proceedings usually published by Springer.

Gesture Workshop 2005 (GW 2005) was organized by VALORIA, at the University of Bretagne Sud (Vannes, France), and was held on Berder Island, Morbihan (France) during May 18-20, 2005. This event, the sixth in a highly successful workshop series, was attended by more than 70 participants from all over the world (13 countries). Like the previous events, GW 2005 aimed to encourage multidisciplinary exchanges by providing an opportunity for participants to share new results, show live demonstrations of their work, and discuss emerging directions on topics broadly covering the different aspects of gesture. The very special area where the workshop took place (a small island in the Gulf of Morbihan) provided an occasion for lively discussions and establishment of future collaboration on research centered on gesture as a means of communication. A large number of high-quality submissions was received, which made GW 2005 a great event for both industrial and research communities interested in gesture-based models relevant to human–computer interaction and simulation.

This book is a selection of revised papers presented at Gesture Workshop 2005. Containing 24 long papers and 14 short papers, it offers a wide overview of the most recent results and work in progress related to gesture-based communication. Two contributions on major topics of interest are included from two invited speakers. The contribution from Jean-Louis Vercher (Movement and Perception Lab., Marseille, France) is concerned with fundamental issues of biological motion, and their link with the perception and the synthesis of realistic motion. The contribution from Ronan Boulic et al. (EPFL, Switzerland) highlights the potential of some well-known computer animation methods for motion synthesis. The book covers eight sections of reviewed papers relative to the following themes:

- Human perception and production of gesture
- Sign language representation
- Sign language recognition
- Vision-based gesture recognition
- Gesture analysis
- Gesture synthesis
- Gesture and music
- Gesture interaction in multimodal systems

Under the focus of gesture in Human-Computer Interaction and Simulation, the book encompasses all aspects of gesture studies in emerging research fields. Two sections are devoted to sign language representation and recognition. Pertinent features extracted from captured gestures (signal, image) are used for processing, segmentation, recognition or synthesis of gestures. These topics concern at least three sections of the book. Different kinds of applications are considered, including for example expressive conversational agents, gesture interaction in multimodal systems, and gesture for music and performing arts.

The workshop was supported by the University of Bretagne Sud (France), the French Ministry of Research, the *Conseil Régional de Bretagne* and the *Conseil Général du Morbihan*: we are very grateful for their generous financial support. GW 2005 also received some financial support from COST-European Science Foundation. In particular, the Cost287-ConGAS action, mainly concerned with Gesture Controlled Audio Systems, was strongly represented within the workshop, and we are grateful to the delegates for their contribution to the event and the book. Thanks also to France Telecom R&D (a French telecommunication society) which generously contributed to the sponsoring of GW 2005, and participated in the forum by presenting very relevant demonstrations.

We would also like to express our thanks to the local Organizing Committee (Sylviane Boisadan, Alexis Héloir, Gildas Ménier, Elisabeth Le Saux, Joël Révault, Pierre-François Marteau) as well as Gersan Moguérou for webmastering the GW2005 Internet site. We are also grateful to the university staff and the PhD students from VALORIA who helped in the organization of the workshop.

Finally, the editors are thankful to the authors of the papers, as well as the international reviewers. As a result of their work, this volume will serve as an up-to-date reference for researchers in all the related disciplines.

December 2005   Sylvie Gibet
                                                    Nicolas Courty
                                           Jean-François Kamp

Reviewers

Table of Contents

Sign Language Recognition

Long Papers

Short Papers

Vision-Based Gesture Recognition

Long Papers

Short Papers

Gesture Analysis

Long Papers

Short Papers

Gesture Synthesis

Invited Paper

Long Papers

Short Papers

Gesture and Music

Long Papers

Gesture Interaction in Multimodal Systems

Long Papers

Short Papers

Perception and Synthesis of Biologically Plausible Motion: From Human Physiology to Virtual Reality

Jean-Louis Vercher

UMR CNRS 6152 Mouvement et Perception, Institut Fédératif de Recherche
Etienne-Jules Marey, Université de la Méditerranée Marseille, France
`vercher@laps.univ-mrs.fr`

Abstract. To model and simulate human gesture is a challenge which takes benefit from a close collaboration between scientists from several fields: psychology, physiology, biomechanics, cognitive and computer sciences, etc. As an a priori requirement, we need to better understand the so-called laws of biological motions, established all along the 20th century. When modelled and used to animate artificial creature, these laws makes these creatures (either virtual or robotic) move in a much more realistic, life-like, fashion.

1 Introduction

A virtual reality (VR) system is expected to provide realistic representations of objects. The realistic character applies at least as much to the behaviour of the objects as to their aspect [1]. A moving object must particularly comply with certain rules: at first, the laws of physics of course, and in particular those of the Galilean kinematics and Newtonian dynamics. The compliance with these rules confers realistic properties to the environment (i.e. gravity) as to the objects (inertia, surface properties, constitution, etc). In the particular case where the animated object corresponds to a living being (animal, human) or supposed such, additional rules are essential, to obtain that the simulated item is perceived as being alive. Indeed, many studies in the field of psychology of perception revealed the existence of biological "signatures" in the movement of the living beings. The existence of these signatures in representations of moving objects, not only are enough "to animate" (within the meaning of "giving life to") the objects, but are essential to allow their recognition as products of a biological activity. All along this paper, we will review the principal studies related to the perception of biological motion, we will see how the designers of virtual reality applications, as those of multi-media and cinema industries take advantage from this knowledge, and we will see finally how virtual reality can help perception psychology to better understand the phenomena which gives the character ALIVE to animated symbolic systems.

2 The Perception of Biological Motion

Studies in psychology and neurophysiology obviously show that the movements of a human body can be easily recognized and identified in their biological, living nature.

The literature on the perception of the so-called biological motion is abundant and varied. This literature really finds its source in the seminal work of Johansson [2], who characterized biological motion as referring to the ambulatory patterns of bipeds and terrestrial quadrupeds.

2.1   Characteristic Points Are Worth a Complex Picture: Johansson

Johansson [2] filmed in total darkness walkers with, as only visible elements, some lights attached to the main joints of the body (Fig. 1). He showed that such points, moving on a uniform background, were perceived by observers as indicating human movements, in the absence of any other visual index. At the end of the 19th century, Muybridge and Marey had already, in an implicit way, used this faculty of our brain to reconstitute the complex gesture from a finished number of points (connected or not between them by straight lines).
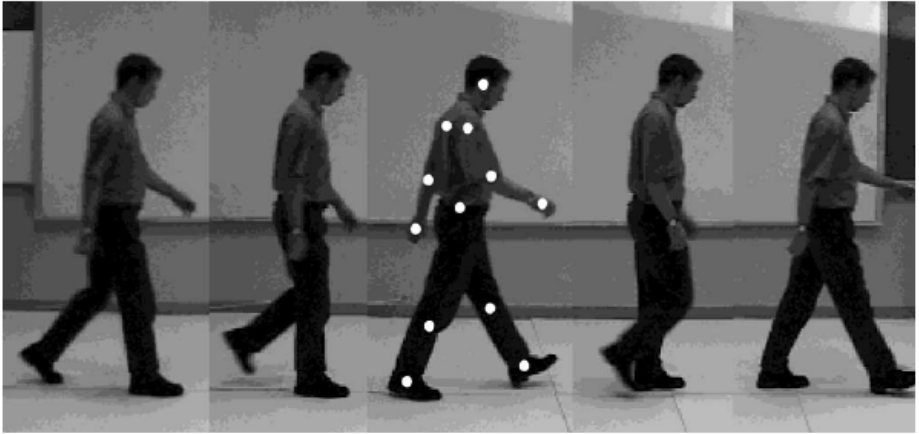
Fig. 1. Example of a walker on which markers are placed on specific joint

Johansson's experiment has been frequently reproduced and confirmed, and has become a traditional chapter of perception psychology. When the filmed people dance, the observer can eventually identify the type of dance [3], and to distinguish the men from the women [4]. This capability to perceive the human or biological character from sparse indices is innate or almost: it has been observed in children as early as 4 to 6 months after birth [5]. The biological nature of movement may be identified even when the group of dots corresponds to the movement of an animal [6], it thus does not act of a specific effect. It even seems that we have the capability to infer the shape and the physical properties of the body from a simple group of moving dots moving. Many visual demonstrations of this effect can be observed online [7, 8].

Most of the studies related to the perception of biological motion do not carry in them any attempt at explanation: they do nothing but show-up the existence of the phenomenon, and in particular they highlight the need for maintaining the characteristics kinematics of the points, whatever they are. Some attempts at modelling of this perceptive capacity were however proposed, in order to explain how the three-dimensional

structure of the movement of the limbs of an animal could be calculated starting from the two-dimensional movements of some markers projected in the plan of the image [9]. At least two of these attempts deserve to be known.

2.2 Paolo Viviani and the Motor Theory of Perception

The determination of invariant characteristics of movement constitutes a millstone of crucial importance to the understanding of the fundamental principles of organization of biological motor control, concerning in particular the role of the central nervous system (CNS). Although his work was not directly related to the problems of the perception of biological motion (but rather to automatic signature recognition), Viviani contributed to the comprehension of the phenomenon. Seeking to identify an invariant in the morphogenesis of writing, he demonstrated the existence of a non-linear relation linking the angular velocity of the hand to the trajectory curvature [10, 11]:

$$a(t)=kc(t)^{2/3} \tag{1}$$

This relation, extremely robust (the human gesture cannot violate it) strongly conditions our perception. Its non-observance leads an observer to confuse the shape of the trajectory: a circle becomes an ellipse and vice and versa. When for example one observes a luminous point moving on an elliptical path according to kinematics corresponding to a circle (constant angular speed), the observer perceives the point as moving along a circle. The vision is not the only sensorial modality concerned: passive movements of the hand induced by a computer via a robot are perceived correctly only if the trajectory is in conformity with that produced by an active movement. This law affects also the eye movements: the trajectory and the performance of visual tracking of a luminous moving point differ if the movement of this the point does not respect the law [12]. According to Viviani [13], the phenomenon finds its origin in the motor theory of perception: our perception of the movement is determined by our way to move and to act.

The power 2/3 law is considered as being a fundamental constraint of the CNS on the formation of the trajectories of the end-point of the gesture (e.g. the hand), in particular when performing rhythmic movements. This law also appears for more complex movements, concerning the whole body, as in locomotion [14, 15]. Confronted to a corpus of convergent experimental data, the power 2/3 law is regarded as an invariant of the trajectory of biological movements, impossible to circumvent and is often used as a criterion of evaluation of animated models [16, 17].

2.3 Local or Global Process: Maggie Shiffrar

The human body may be considered as a chain of rigid, articulated elements, giving to the body a non-rigid aspect. Shiffrar [18] attempted to determine how we can perceive a body as moving, and in particular how the visual system can integrate multiple segmental information on the moving body in order to perceive this movement as a single and continuous event. The assumption is that movement (animation) allows the establishment of a bond (rigid or not) between the various points. Beyond the perception of biological motion, this concept can be generalized to the perception of any
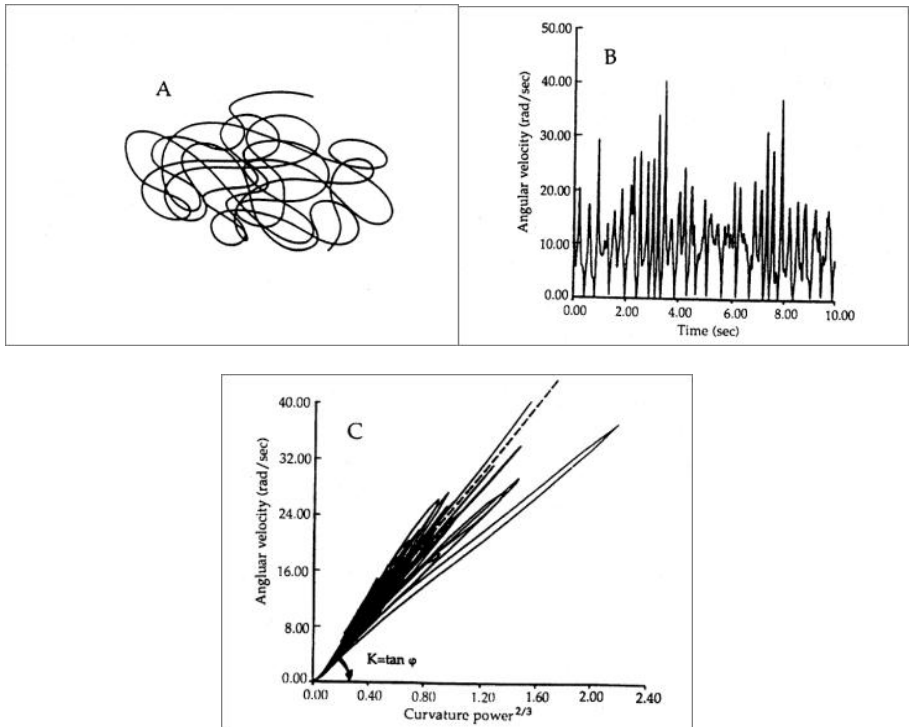
Fig. 2. Neither the trajectory of a graphic gesture (A) nor the time-course of angular velocity (B) reveal an invariant in the production of the gesture. On the other hand, the angular velocity is systematically proportional to the curvature of the trajectory raised at power 2/3 (C).

physical object and its identification by the visual system [19]. One of the questions tackled by the studies on perception of biological motion relates to the primary level of analysis, global or local (does each point count, or is it rather the general pattern which is relevant?). Shiffrar pleads for a global analysis: a group of dots representing a walker is not anymore recognized as such if it is reversed from top to bottom [20]; on the other hand, the walker is identified even if the characteristic points are drowned in a background of random dots [21]. It should finally be considered that more probably neither the shape (the space distribution) of the group of dots, nor the characteristics of the movement of the points are enough alone to define or categorize human movement. The capability of the visual system to extract a human form from motion is based on the space-time integration of the indices of BOTH form AND movement and must thus be regarded as a phenomenon of both local and global nature [18].

2.4   From Perception to Memory of Biological Motion: Perceptive Anticipation

This spatio-temporal integration does not only affect perception but also memory. Under certain conditions, our memory of the final position of a moving target which

is abruptly stopped is distorted in the direction of the represented movement [22]. In the same way, our memory of a static view of a moving object or character is biased in the direction of the movement [23]. This phenomenon, called "representational momentum" attests of the "interiorisation" of the physical principle of inertia. It has been recently shown that the perception of complex biological movements is also affected by this phenomenon. Thus, even in case of disruption of a visual stimulation corresponding to a complex gesture, the perception of the event and its dynamics remain, resulting in a bias of the memory of the final posture, shifted in the direction of the movement [24].

The identification of actions and human body postures is a major task of our perceptive system, which depends on the point of view adopted by the observer. Indeed, visual recognition of a furtive human posture (presented during 60 ms) is facilitated by the previous presentation of identical postures, but only if these previous presentations are from close points of view [25]. In the same way, it is easier to assess the biological realism of a posture if the movement preceding this posture is also presented to the observer. Thus, one can anticipate the postures to result from a subject's movement, facilitating the identification of these postures [26]. Other studies on perceptive anticipation abound in this direction: it is enough for an observer to perceive the beginning of a gesture (i.e. a writing sequence) to correctly predict the nature of the incoming movements, and even if the produced series of letters do not form words [2].

2.5   Neural Substrates

Perception of biological motion is not a human exclusive capability: animals, and in particular monkeys or pigeons are also sensitive. This allowed, through a number of electrophysiological studies in primates (cell recording), to determine the concerned cortical zones. Oram and Perrett [28] showed that neurons of the temporal superior area respond to this kind of specific stimuli. Newsome and Paré [29] showed that monkeys can detect the direction of a movement when the level of coherence is as low as 1% or 2%. Destruction of a specific cortical zone (MT) increased this threshold of coherence to 10% or 20%.

Brain functional imaging now makes it possible to identify, on humans, zones of the brain specifically activated during perception and recognition of biological motion, and especially to determine the networks involved. The supero-temporal sulcus seems to play a particular role in the perception of biological motion [30] (Fig. 3). These studies made it possible to highlight the implication of a large number of zones, including of course those directly concerned with vision (in particular the lingual gyrus [31]); and the superior occipital gyrus [32] but also perceptive processes in general (temporal and parietal cortices), as well as other parts of the brain generally concerned with the control of movement (premotor cortex, lateral cerebellum), thus confirming the close link between the generation of movement and its perception [33].

Thus, for example, experiments of neuroimagery carried out in monkeys [34] and on humans [35, 36] showed that neurons known as "mirror neurons", located in the lower part of the premotor cortex are active both when one observes somebody
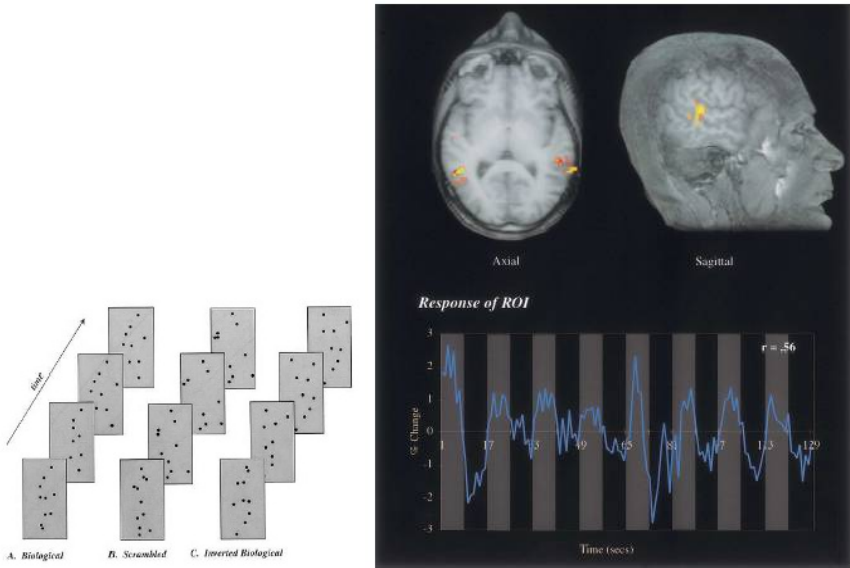
Fig. 3. The activity in specific zones of the brain (here the superior temporal sulcus) is modulated according to the presentation of a biologically compatible stimulus vs. a non compatible (scrambled or inverted) [30]

performing a gesture or when the subject is performing himself the gesture. This network would have a fundamental role in the processes of training by imitation and social communication [37]. These studies and those concerning the role of the ventral and dorsal flow of visual information in motion perception are clearly related.

3 Does Virtual Reality Have to Take into Account the Laws of Biological Motion?

At this point, I would like to emphasise the need, for people involved in VR, to know the main theories of perception and movement control, proposed throughout the 20th century by the psychologists and the physiologists. These theories are in three categories:

- Cognitive theories: they consider perception as a particular form of inferential process, although mainly unconscious. This point of view finds its origin in Helmoltz' work.
- Ecological theories, assuming that we (actors) have a privileged relation with the environment, and that we are directly in interaction with this latter, as proposed by Gibson [38].
- Computational theories, assuming that perception is primarily the result of data processing. Marr is one of the pioneers of this approach.

It is necessary at this stage to distinguish motion perception from perception of biological motion. For Davis [39], at least two reasons justify the need for motion perception:

− To give a sense to the world whereas we move;
− To understand the objects moving around us.

It should well be understood that, in a VR context, the distinction between the two cases (movements of the self body in a stable environment, and movement of the scene around a motionless observer) is not trivial, in particular for the observer/actor himself, and the problem is even worse in the general case, when the observer moves among objects themselves in movement. In a finer analysis, Nakayama [40] suggested seven reasons for which perception of movement is an important issue. It allows:

− To derive the 3rd dimension from 2D information (motion parallax);
− To calculate the time before contact;
− To distinguish an object from the background;
− To obtain information related to the execution of our own movements;
− To stimulate the ocular movements;
− To understand patterns;
− To perceive the moving objects.

Warren [41] goes still further and stresses the role of the relation between action and perception, all together being regarded as a whole. These elements emphasise the importance of the detection of movement for the success of our actions. There is thus little doubt that movement in general helps us to understand the world and the objects within it. It is frequent to note that animation gives sense to pictures, which, motionless, are little informative. It is particularly true with wire-frame pictures of 3D objects: even a large number of lines in the graphic scene does not make possible to disambiguate spatial ambiguities (i.e. the Necker cube). When these images are animated and the observer is allowed to change the point of view or to make the objects turn, these objects suddenly take "life" and the 3rd dimension appears without effort. Movement parallax is indeed one of the most powerful indices of depth.

Obstacles to a correct perception of movement are however numerous in a context of VR. Let us quote among others:

− When frame frequency (sampling) is too low to maintain a sensation of movement continuity;
− When latency is to high between user's movement and the sensory feedback;
− When incoherence between the various sensorial feedbacks (visual, haptic, sound) dramatically degrades perception.

Within this framework, the processing or the simulation of biological motion is indeed a particular case. Multimedia industries (games, cinema) understood it particularly well, even if the implementation is largely based of empirical methods. The guiding principle of this implementation, characters animation (virtual actors or avatars) is based on techniques of movement recording (motion capture) and modelling (dynamic data morphing) of the kinematical data obtained in order to adjust these data to the geometrical properties of the avatar [42, 43]. The Web site of these authors

illustrates the technique [44]. These techniques, requiring off-line treatment, are not easily transferable to virtual reality, excepted when the goal is to animate characters according to a pre-established scenario. It is also possible to model certain aspects of the biological movements in order to take these laws into account during the animation of the avatars. The power 2/3 law is of these aspects [45]. The recent availability of motion capture systems and real-time animation techniques makes it possible to exceed these limitations [46]. Avatar animation of the main actor (the user of the VR system) most of the time complies with the rules of the biological motion, whatever they are, since this animation is done on the basis of real-time motion capture of a real biological being, the user himself. It remains important to check that it is so. The suggested procedure is as follows:

– To reduce the animated character to his simpler expression, ideally a characteristic group of dots, in order to remove any source of contextual information;
– To apply a procedure of evaluation of the biological movement by naïve observers. That proposed initially by Johansson [2] proved to be satisfactory and is sufficiently validated.

4   Will Virtual Reality Make It Possible to Better Understand the Perception of the Biological Motion?

In a majority of psychological or physiological studies on perception of biological motion, the stimulus is often limited to a set of dots supposedly attached to the joints of a person. In spite of this drastic information degradation, the human visual system organizes the dots pattern in an undeniable percept of a biological creature.

Various techniques were used in order to generate the group of dots, from video recording to pure simulation, including motion capture. If the first leaves little possibility of deterioration of the signal, a combination of the two last makes it possible to obtain data realistic enough and to handle them on computer, with the aim of identifying pertinent information [47]. Let us remember that this information, hidden in the space-time structure of the image, is still not clearly identified.

It is clear that virtual reality can provide a much richer, flexible and general-purpose tool for the psychological examination of motion perception, but also for physiological studies, by the conjunction of VR techniques and cerebral functional imaging [48]. This could not only rely on an improved picture control, but also on the interactive and immersive nature inherent to VR. Virtual reality is already largely used for the study of motion perception in general, and in particular for the perception of movements of the self body: real-time animation of an avatar by a subject exposed to a vection stimulus makes it possible on this subject to give a quantifiable image, other that verbal, of its illusory movement [49]. Virtual reality constitutes also an extremely promising tool to study space orientation, by its capacity to generate complex and realistic environments (i.e. urban). It makes it possible to generate experimentally controlled conflicts [50].

In a more general way, virtual reality offers an opportunity to create synthetic environments with a high number of variables influencing our behaviour, and these environments can be easily and precisely controlled; virtual reality allows the creation of dynamic 3D views, providing to the user (or the subject) visual information rather

close to that obtained in a real scene. In this direction, VR goes much further that the groups of dots, random or not, generally used in psychological and physiological experiments on vision. In this sense, VR opens the door towards a new approach of experimental study of perception of movement. This new approach takes really its rise only by the association of VR specialists on the one hand and of integrative Neurosciences scientists on the other. Many associations of this kind, especially in the United States (such as for example VENlab at Providence [51, 52], or Loomis' laboratory in Santa Barbara [53], but also in France (for example at the Marey Institute, Marseilles) showed the potential richness of this multi-field approach.

5   Conclusion

From a tremendous amount of scientific literature existing on the problem of perception of biological motion, we can retain the following points:

− Some invariants exist in the kinematics of the movement of the animals which are identifiable by an observer as fundamental characteristics of life.
− These invariants are to be sought both in the kinematics of the individual points and in the spatio-temporal organization of the pattern (e.g. phase ratios between all the points). These invariants concern the local and the global level.
− This capability to perceive movement of biological origin is innate (it does not need to be learned) and cannot be unlearned. It cannot be isolated from our perception of the world and our way of acting on it. Any change, even light, of these specific kinematics deteriorates dramatically our perceptive capabilities.
− This capability of living beings to perceive a biological character in movement must be taken into account at the time of the simulation of virtual worlds, competitively with other aspects of simulation which could prove, in certain cases, less rich in "useful" information for the user, while being, sometimes, considerably more expensive in resources and time.
− Virtual reality offers an environment potentially rich for the study of perception, and more particularly the links between movement (or action) and perception. The community involved the study of perception and action understood it soon, but the community is far from exploiting all the potential richness.

In conclusion, when one aims at simulating living beings acting in a virtual environment, it appears necessary that the conceptors of VR systems have in mind a certain level of comprehension of the mechanisms of perception, not only visual, but concerning all the senses involved in the particular VR application. They must be able to exploit the variables as well as the invariants identified in previous physiological and psychological studies, this in order to generate and to transmit information in its most adapted form, according to the desired task. In addition, psychologists, who often regard VR as a tool well adapted to their experiments, are likely to pass beside fundamental questions concerning VR (i.e. perception in immersion, the nature of the state of immersion...). It is extremely difficult, today, to gauge which is exactly the level "of intercommunicability" between the two communities, but one can hope that it will be reinforced in the incoming years, for the benefit of the two communities. One of the means to go towards this goal, if not to reach it, is to gather, on precise

scientific objectives, multi-disciplinary project-teams bringing together data processing specialists, computer scientists, cognitivists, physiologists and psychologists of perception and human movement.

References

1. Fuchs, P., Moreau, G., Papin, J.P. : Le traité de la Réalité Virtuelle ; Presses de l'Ecole des Mines de Paris, Paris (2001)
2. Johansson, G.: Visual perception of biological motion and a model for its analysis. Perception and Psychophysics 14 (1973) 201-211
3. Dittrich, W.H., Troscianko, T., Lea, S.E., Morgan, D.: Perception of emotion from dynamic point-light displays represented in dance. Perception 25 (1996) 727-738
4. Mather, G., Murdoch, L.: Gender discrimination in biological motion displays based on dynamic cues. Proc. Royal Soc. London B. Biol. Sci. 258 (1994) 273-279
5. Fox, R., McDaniel, C.: The perception of biological motion by human infants. Science 218 (1982) 486-487
6. Mather, G., West, S. Recognition of animal locomotion from dynamic point-light displays. Perception 22 (1993) 759-766
7. http://zeus.rutgers.edu/~feher/kutya_e/example1.html
8. http://www.psy.vanderbilt.edu/faculty/blake/BM/BioMot.html
9. Hoffman, D.D., Flinchbaugh, B.E.: The interpretation of biological motion. Biol; Cybern. 42 (1982) 195-204
10. Viviani, P., Terzuolo, C.: Trajectory determines movement dynamics. Neuroscience 7 (1982) 431-437
11. Lacquaniti, F., Terzuolo, C., Viviani, P.: The law relating the kinematic and figural aspects of drawing movements. Acta Psychologica 54 (1983) 115-130
12. De'Sperati, C. Viviani, P.: The relationship between curvature and velocity in two-dimensional smooth pursuit eye movements. J. Neuroscience 17 (1997) 3932-3945
13. Viviani, P.: Motor perceptual interactions: the evolution of an idea. In: "Cognitive Science in Europe: issues and trends", Piattelli Palmarini, M., ed. Golem (1990) 11-39
14. Vieilledent, S., Kerlirzin, Y., Dalbera, S., Berthoz, A.: Relationship between velocity and curvature of a human locomotor trajectory. Neurosci. Lett. 305 (2001) 65-69
15. Ivanenko, Y.P., Grasso, R., Macellari, V., Lacquaniti, F. Two-thirds power law in human locomotion : role of ground contact forces. NeuroReport 13 (2002) 1171-1174
16. Viviani, P., Flash, T.: Minimum-jerk, two-thirds power law, and isochrony: converging approaches to movement planning. J. Exp. Psychol.: Hum. Perc. Perf. 21 (1995) 32-53
17. Harris, C.M., Wolpert, D.M.: Signal dependent noise determines movement planning. Nature 394 (1998) 780-784
18. Shiffrar, M., Lichtey, L. Heptulla Chatterjee, S.: The perception of biological motion across apertures. Perception and Psychophysics 59 (1997) 51-59
19. Kourtzi, Z., Shiffrar, M.: Visual representation of malleable and rigid objects that deform as they rotate. J. Exp. Psychol.: Hum. Perc. Perf. 27 (2001) 335-355
20. Sumi, S.: Upside-down presentation of the Johansson moving light-spot pattern. Perception, 13 (1984) 283-286
21. Pinto, J., Shiffrar, M.: Subconfigurations of the human form in the perception of biological motion displays. Acta Psychol. 102 (1999) 293-318
22. Hubbard, T.L.: Environmental invariants in the representation of motion. Psychon Bull & Rev. 2 (1995) 322-338

23. Freyd, J.J.: The mental representation of movement when static stimuli are viewed. Percep & Psychophys. 33 (1983) 575-581
24. Amorim, M.-A., Wexler, M.: Perception des mouvements biologiques complexes et principes physiques invariants. In Actes du IXème Congrès International des Chercheurs en Activités Physiques et Sportives, Valence, 1-3 novembre 2001
25. Daems, A, Verfaillie, K.: Viewpoint-dependent priming effects in the perception of human actions and body postures. Vis Cogn, 6 (1999) 665-693.
26. Verfaillie, K., Daems, A.: Representing and anticipating human actions in vision. Vis Cogn. 9 (2002) 217-232
27. Orliaguet, J.P., Kandel, S., Boe, L.J.: Visual perception of motor anticipation in cursive handwriting: influence of spatial and movement information on the prediction of forthcoming letters. Perception 26 (1997) 905-912
28. Oram M.W., D.I. Perrett, D.I.: Neural Networks 7 (1994) 945–972
29. Newsome W.T., Pare, E.B.: A selective impairment of motion perception following lesions of the middle temporal visual area (MT). J Neurosci. 6 (1988) 2201-11
30. Grossman E.D., Blake R.: Brain activity evoked by inverted and imagined biological motion. Vision Research 41 (2001) 1475-1482
31. Servos P, Osu R, Santi A, Kawato M.: The neural substrates of biological motion perception: an fMRI study. Cerebral Cortex 7 (2002) 772-82
32. Paradis, A.L., Cornilleau-Peres, V., Droulez, J., Van de Moortele, P.F., Lobel, E., Berthoz, A., Le Bihan, D., Poline, J.B.: Visual Perception of motion and 3D structure from motion : an fMRI study. Cerebral Cortex 10 (2000) 772-783
33. Vaina, L.M., Solomon, J., Chowdhury, S., Sinha, P., Belliveau, J.W.: Functional neuroanatomy of biological motion perception in humans. Proc. Nat. Acad. Sci. 98 (2001) 11656-11661
34. Rizzolatti, G., Fadiga, L., Matelli, M., Bettinardi, V., Paulesu, E., Perani, D., Fazio, F.: Localization of grasp representations in humans by PET: 1. Observation versus execution. Experimental Brain Research 111 (1996) 246–252
35. Nishitani, N., Hari, R.: Dynamics of cortical representation for action. Proceedings of the National Academy of Science 97 (2000) 913–918
36. Buccino, G., Binkofski, F., Fink, G.R., Fadiga, L., Fogassi, L., Gallese, V., Seitz, R. J., Zilles, K., Rizzolatti, G., Freund, H.-J. (2001) Action observation activates premotor and parietal areas in a somatotopic manner: an fMRI study. European Journal of Neuroscience, 13, 400-404.
37. Iacoboni, M., Woods, R.P., Brass, M., Bekkering, H., Mazziotta, J. C., Rizzolatti, G.: Cortical mechanisms of human imitation. Science 286 (1999) 2526-2528
38. Gibson, J.J.: The ecological approach to visual perception. Houghton Mifflin, Boston (1979)
39. Davis, S.D.: The design of virtual environments with particular reference to VRML. Advisor Group on Computer Graphics, SIMA report (1996)
40. Nakayama, K.: Biological image motion processing: a review. Vision Research 252 (1985) 625-660
41. Warren, W.H.: Self-motion: visual perception and visual control. In: "Perception of space and motion", W. Epstein & S. Rogers, eds. Academic Press, San Diego (1995) 263-325
42. Rigotti, C., Cerveri, P., Andreoni, G., Pedotti, A., Ferrigno, G.: Modeling and driving a reduced human mannequin through motion captured date: a neural network approach. IEEE Trans. Sys. Man Cybern. 31 (2001) 187-193
43. Hodgins, J.K., O'brien, J.F., Tumblin, J.: Perception of human motion with different geometric models. IEEE Trans. Visual. Comput. graph. 4 (1998) 307-316

44. http://www.cc.gatech.edu/gvu/animation
45. Soechting, J.F., Terzuolo, C.: An algorithm for the generation of curvilinear wrist motion in an arbitrary plane in three-dimensional space. Neuroscience 19 (1986) 1393-1405
46. Tang, W., Cavazza, M., Moutain, D., Earnshaw, R.: A constrained inverse kinematic technique for real-time motion capture animation. Visual Computer 15 (1999) 413-425
47. Dekeyser, M., Verfaillie, K., Vanrie, J.: Creating stimuli for the study of biological-motion perception. Behav. Res. Methods Instrum. Comput. 34 (2002) 375-382
48. Grèzes, J., Fonlupt, P., Bertenthal, B., Delon-Martin, C., Segebarth, C. Decety, J.: Does perception of biological motion rely on specific brain regions? NeuroImage 13 (2001) 775-785
49. Parker, D.E., Phillips, J.O.: Self-motion perception: assessment by real-time computer-generated animations. Applied Ergonomics 32 (2001) 31-38
50. Lambrey, S., Viaud-Delmon, I., Berthoz, A.: Influence of a sensorimotor conflict on th ememorization of a path traveled in virtual reality. Cognitive Brain Research 14 (2002) 177-186
51. Tarr, M.J., Warren, W.H.: Virtual reality in behavioral neuroscience and beyond. Nature Neuroscience 5 (2002) 1089-1092
52. http://www.cog.brown.edu/Research/ven_lab/
53. Loomis, J.M., Blascovitch, J.J., Beall, A.C.: Immersive virtual environment technology as a basic research tool in psychology. Behav. Res. Meth. Instr. Comput. 31 (1999) 557-564

Temporal Measures of Hand and Speech Coordination During French Cued Speech Production

Virginie Attina, Marie-Agnès Cathiard, and Denis Beautemps

Institut de la Communication Parlée, UMR CNRS 5009,
INPG, 46 avenue Félix Viallet 38031 Grenoble, France
{attina, cathiard, beautemps}@icp.inpg.fr

Abstract. Cued Speech is an efficient method that allows orally educated deaf people to perceive a complete oral message through the visual channel. Using this system, speakers can clarify what they say with the complement of hand cues near the face; similar lip shapes are disambiguated by the addition of a manual cue. In this context, Cued Speech represents a unique system that closely links hand movements and speech since it is based on spoken language. In a previous study, we investigated the temporal organization of French Cued Speech production for a single cueing talker. A specific pattern of coordination was found: the hand anticipates the lips and speech sounds. In the present study, we investigated the cueing behavior of three additional professional cueing talkers. The same pattern of hand cues anticipation was found. Results are discussed with respect to inter-subject variability. A general pattern of coordination is proposed.

1   Introduction

It is well known that for many hearing-impaired people, lipreading is the key to communicating with others in everyday situations. Unfortunately visual interpretation of lip and mouth gestures alone does not allow the totality of the oral message to be distinguished due to the ambiguity of the visual lip shapes. This leads to a general problem of speech perception for deaf people.

Cued Speech (CS) is a visual communication system that uses handshapes placed in different positions near the face in combination with the natural mouth movements of speech to make the sounds of spoken language look different from each other [1]. Adapted to more than 56 languages [2], it represents an effective method that enhances speech perception for the deaf. With this system, speakers while talking execute a series of hand and finger gestures near the face closely related to what they are pronouncing; the hand, with the back facing the perceiver, constitutes a cue which uniquely determines a phoneme when associated to a lip shape. A manual cue in this system is made up of two components: the shape of the hand (finger configuration) and the position of the hand near the face. Hand shapes are designed to distinguish among consonants and hand positions among vowels. The manual cues are defined so that the phonemes that are

visually similar on the lips are coded by perceptually distinctive manual cues, with a manual cue corresponding to a subgroup of visually contrastive phonemes. Thus manual and labial information complement each other; given alone by the hand or the lips, the information is ambiguous. Figure 1 illustrates the manual cues for French phonemes (*Langue française Parlée Complétée*, LPC; or *French Cued Speech*, FCS).

This system is based on a CV (Consonant-Vowel) manual resyllabification of speech. To code a CV syllable, one simultaneously forms the specific finger configuration for the consonant C and moves the hand to the specific position corresponding to the vowel V. In case of isolated consonants or vowels, one uses the appropriate hand shapes for the consonants at the side position and uses the corresponding positions for the vowels with hand shape 5 (see Fig. 1).
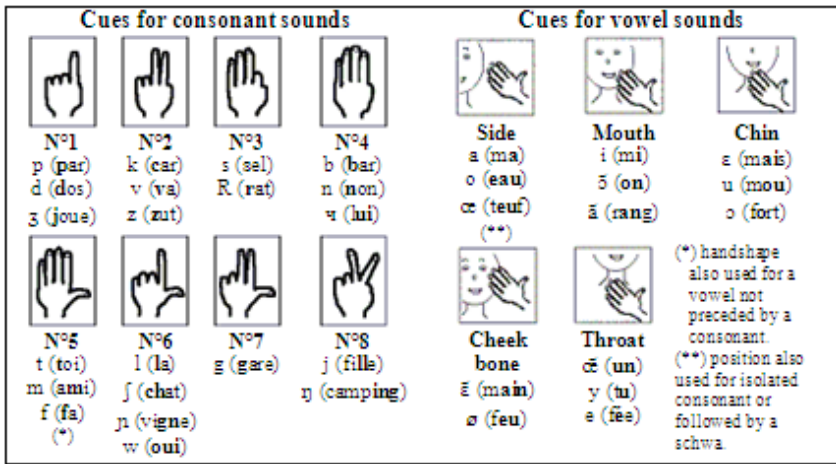
Fig. 1. Manual cues for French vowels and consonants

Seeing manual cues – the hand shape at a specific position – associated to lip shapes allows the deaf cue perceiver to identify through vision only the exact speech message transmitted. Many studies have shown evidence for the effectiveness of Cued Speech for visual phoneme perception: additional manual cues can strongly improve speech perception by deaf people from 30% to more than 90% of accurate perception ([3], [4]). This can result in a noticeable improvement in lipreading acquisition and oral speech decoding by deaf children using this method. Several studies have shown evidence for the value of using CS with deaf children particularly at early ages to acquire complete phonological representations (for a review see [5]).

The remarkable effectiveness of CS for speech perception gives some evidence that during cued speech production, hand and lip movements are organized in a coordinated manner. They are tightly linked by definition, since the shape and

the position of the manual cue depends on speech. While hand and face gestures coordination appears to be the key factor in this system (this was also emphasized in studies with technological purposes like automatic CS synthesis [6], [7]), very little is known about Cued Speech production, i.e. the organization and timing of the hand and the lips in relation to speech sounds during speech cueing. How do the manual gestures occur with speech vocalizations in the course of this "artificial" syllabic system? What about the cue-timing strategies that make this system so efficient for speech perception? In order to give some answers to these questions, we previously investigated the temporal organization of French Cued Speech (FCS) in the performance of one professional cueing speaker [8]. A specific pattern of coordination was found: the hand anticipates the lips and speech sounds. More precisely, for cueing a CV syllable, the temporal pattern observed was: (1) the displacement of the hand towards the target position could begin more than 200 ms before the consonantal acoustic onset of the CV syllable. This implied that the gesture began in fact during the preceding syllable, i.e. during the preceding vowel. (2) The hand target was attained around the acoustic onset of the consonant (during the first part of the consonant). (3) The hand target position was therefore reached largely before the corresponding vocalic lip target (on average 172 to 256 ms before the vowel lip target). (4) Finally, the hand left the position towards the next position (corresponding to the following syllable) during the production of the vowel. The hand shape was entirely formed during the hand transition: the hand shape formation was superimposed on the hand transition from one position to another and did not disturb the manual displacement.

The aim of the present study is to see whether this coordination pattern was subject-dependent or is a general feature of FCS production. We therefore recorded three other professional cueing speakers producing a wide corpus of syllabic sequences in order to investigate the temporal pattern of Cued Speech production across the subjects. The study focused on hand gesture timing during FCS production, so only hand transitions were analyzed. Results are discussed with respect to intra- and inter-speaker variability during the cued syllable production. General observations on Cued Speech organization are proposed.

2   Method

2.1   Subjects

The subjects were three French female speakers (ranging in age from 30 to 45 years) with normal hearing. They were all nationally certified as professional French cueing talkers and were experts in the practice of manual CS (number of years of cueing practice ranged from 4 to 14 years with at least 17 hours of professional cueing per week).

2.2   Corpus

Syllabic sequences decomposed as [C1V1.C1V1.C2V2.C3V1] (S0S1S2S3) were used for the corpus with: the consonants [m] or [b] for C1; {[p], [j]}, {[s], [l]},

{[v], [g]}, or {[b], [m]} respectively for C2 and C3; the vowels [a, i, u, ø, e] for V1 and V2 (excluding the case where V1=V2) (e.g. [ma.ma.be.ma]; see the complete stimulus materials in appendix). Their combination gives a total of 160 sequences involving both hand transitions and finger gestures and exploiting the eight hand shapes and the five positions of FCS code. The analysis focused on the embedded S2 syllable (C2V2), including the transitions from S1 syllable to S2 and S2 to S3, in order to bypass the effects relative to the sequence onset and offset.

2.3   Experimental Procedure

The three recordings were made in a sound-proof booth. The subject was audiovisually recorded by two synchronous cameras at 50 frames per second. One camera was used to film the movement of the hand in the 2-D plane and the other in zoom mode to accurately capture the details of lip movements. The subject worn opaque eyeglasses used as protection against the strong lighting conditions and as a reference point for the different measurements (the displacements of the hand are referenced to a colored mark on one of the lenses of the eyeglasses). The speaker uttered and coded at a normal rate the sequences, which were firstly pronounced by an experimenter.

Hand movements consisted of trajectories from one position to another around the face in the 2-D plane. They were therefore readily measurable in the vertical and horizontal dimensions (x and y coordinates) and to this aim, colored markers were placed on the back of the hand of the speaker in order to automatically video-track hand trajectories in the plane. Lips were painted blue in order to extract the internal lip contours using the video processing system of Institut de la Communication Parlée [9]. This software provides labial cinematic parameters; we selected lip area function as a good articulatory descriptor of the lips [10] since it is the parameter most directly related to vowel acoustics [11]. Thus processing of the video delivered the x and y coordinates of the hand markers and lip area values as a function of time every 20 milliseconds. The acoustic signal was digitized synchronously with the video signal and sampled at 44100 Hz. Thus, at the end of data processing, four synchronous signals versus time were obtained for each sequence: lip area evolution (50 Hz), x and y coordinates of the hand markers (50 Hz) and acoustic signal (44100 Hz). Data extraction is illustrated in Fig. 2, which shows segments of the four signals versus time for the [ma.ma.be.ma] sequence produced by one of the subjects.

The signals for lip area, hand horizontal displacement (x) and hand vertical displacement (y) as a function of time were differentiated and low-pass filtered to obtain time-aligned acceleration traces (not shown on Fig. 2). For the analysis, on each signal, temporal events of interest relative to the syllable under study were manually labeled: the onset of consonant acoustic realization (A1) defined on the acoustic waveform and the spectrogram; the vocalic lip target (L2) defined as the moment the lip movement to form the vowel target ends (corresponding to a peak on the acceleration trace); the onset and offset of hand movement delimiting the manual transition coding the S2 syllable (M1 and M2) and
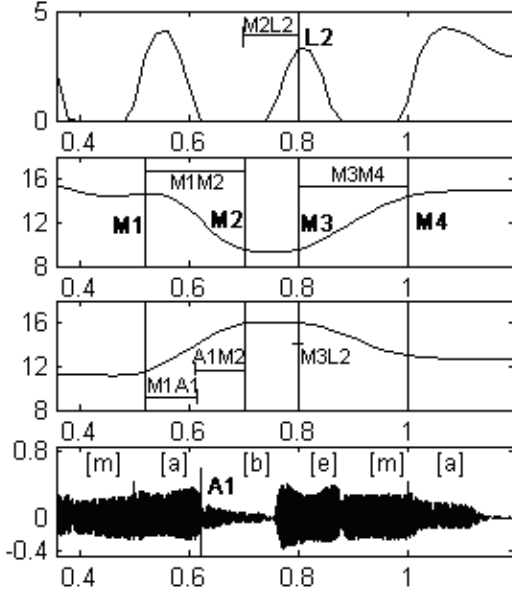
Fig. 2. Signals versus time for [ma.be.ma], part of the [ma.ma.be.ma] sequence of subject 2. From top to bottom: temporal evolution of lip area (cm2), x trajectory of the hand mark (cm), y trajectory of the hand mark (cm) and acoustic signal (s). On each signal, labels used for the analysis are indicated: *L2* (vocalic lip target), *M1* (hand movement onset for [be] syllable), *M2* (hand movement offset for [be] syllable), *M3* (hand movement onset for [ma] syllable), *M4* (hand movement offset for [ma] syllable) and *A1* (acoustic onset of the syllable). See text for the definition of the intervals.

labeled on acceleration and deceleration peaks moments ([12],[13]) and finally the onset and offset of hand movement delimiting the manual transition coding the following syllable S3 (M3 and M4). For more details on data processing, see the description of the method in [8].

2.4   Labels to Production Features

As indicated above, syllable S2 of each [S0S1S2S3] sequence is the critical syllable, i.e. the syllable analyzed. From the acoustic signal, we calculated syllable duration and consonant duration for each sequence. In order to compare the temporal structure of the different signals, some duration intervals were calculated by subtracting the times of cinematic events or acoustic events from one another:

 – M1A1, the interval between the onset of the manual gesture and the acoustic onset of the consonant;
 – A1M2, the interval between the acoustic consonant onset and the offset of the manual gesture;

- M1M2, the interval between the onset and the offset of the manual gesture for S2 vowel;
- M2L2, the interval between the instant the hand position is reached and the moment the lips form the vocalic target;
- M3L2, the interval between the instant the hand leaves the position toward the following position (coding S3) and the vocalic lip target;
- M3M4, the interval between the onset and the offset of the manual gesture for S3 vowel.

The duration of each interval was first computed as the arithmetic difference (for example M1A1=A1−M1 (ms)). Thus, considering a XY temporal interval, a positive value indicates that event X occurs temporally before event Y; conversely a negative value indicates that event X occurs after event Y. The duration of each interval was then quantified as a percentage relative to the duration of the corresponding syllable (%rel). This means that the results will be presented relative to the corresponding temporal information of the acoustic syllable. So a value of 100 indicates that the interval has the same duration of the acoustic CV syllable. And a value smaller than 100 indicates that the interval has a smaller duration than that of the acoustic CV syllable.

3   Results

Results are first presented with milliseconds values: this gives a temporal coordination pattern for hand, lips and speech sounds. Results are then normalized in order to statistically compare cueing behaviors of the three subjects.

3.1   A Temporal Pattern of Coordination

Results in milliseconds are shown in Table 1. First of all, we notice a great similarity in duration for the three subjects. The three cued speech rates are very close: a mean value of 4 Hz calculated from the mean syllable durations was obtained. A one-way analysis of variance (ANOVA) was performed indicating the similarity of the CV syllable duration over the three subjects ($F < 1$). With respect to hand transitions, we notice the proximity of the manual transition durations for each subject: M1M2 and M3M4 intervals are very similar. This result reveals that the rhythm generated by the hand moving from one position to another is rather stable within a subject, whether the hand arrives at or leaves the target position.

With respect to the coordination between hand, lips and sound, we notice from the intervals that the hand gesture is always initiated in advance of the sound for the three subjects: M1A1 interval can vary on average from 143 ms to 153 ms depending on the subject. When looking at the individual items, it should be noted that across all sequences and all subjects, only three items demonstrated a slight delay of the hand over the sound (in fact, closer to synchrony): the anticipatory behavior of the hand thus appears to be a general feature of FCS production. The hand target position is reached during the acoustic production

Table 1. For each of the three subjects, means and standard deviations (into brackets) in milliseconds for all the production features: CV syllable duration, consonant duration, M1M2, M1A1, A1M2, M2L2, M3L2 and M3M4 (See text for details)

Mean duration in ms (std)	Subject 1	Subject 2	Subject 3
CV syllable	252 (41)	253 (45)	258 (56)
Consonant	119 (37)	141 (41)	147 (51)
M1M2	170 (29)	174 (37)	192 (33)
M1A1	153 (56)	145 (56)	143 (50)
A1M2	17 (51)	29 (55)	49 (49)
M2L2	155 (54)	143 (50)	123 (66)
M3L2	9 (73)	-13 (57)	-41 (64)
M3M4	183 (34)	175 (33)	197 (37)

of the consonant: A1M2 interval can vary from 17 ms to 49 ms. This result shows that the hand position is reached just after the acoustic beginning of the consonant, during its first part (the calculation of the corresponding proportions with respect to consonant duration indicates that the manual position is attained at 14% of the consonant duration for subject 1, 21% for subject 2 and 33% for subject 3). With respect to the lips, the hand is placed at the target position well before the vocalic lip target: M2L2 interval can vary on average from 123 ms to 155 ms. Thus the vocalic information delivered by the hand position is always in advance of the one delivered by lip shape. Finally, the hand maintains the target position throughout the production of the consonant and then leaves the position toward the following position around the vocalic lip target realization: indeed, M3L2 interval can vary from −41 ms to 9 ms depending on the subject. This interval demonstrated more variability over the subjects, but what emerges is that the hand leaves the position during the production of the acoustic vowel.

To sum up, the following pattern for FCS production can be built from the temporal intervals obtained: the hand begins its movement before the acoustic onset of the syllable (M1A1 from 143 to 153 ms) and attains its position at the beginning of the syllable (A1M2 from 17 to 49 ms), well before the vocalic lip target (M2L2 from 123 to 155 ms). The hand then leaves the position towards the next position during the vowel. This pattern of coordination appears to be organized exactly the same way as the one obtained previously for a single subject [8]. So the anticipatory behavior of the hand over the lips and the sound appears to be a general feature of FCS production.

3.2   Inter-subject Comparison

With respect to this pattern of coordination, we statistically compared results from the three subjects. To normalize their results, the temporal intervals were quantified as percentages relative to the CV syllable duration of each item (%rel). Results obtained are shown in Table 2.

The three subjects seem to have a quite similar temporal pattern of coordination, as was emphasized before. All three subjects show an advance of the hand

Table 2. For each of the three subjects, means and standard deviations (into brackets) of the production features quantified as percentages relatively to the CV syllable duration (%rel): M1A1, A1M2, M2L2 and M3L2 (See text for details)

Mean duration in ms (std)	Subject 1	Subject 2	Subject 3
M1A1	63 (28)	61 (29)	60 (27)
A1M2	6 (21)	10 (22)	18 (19)
M2L2	62 (21)	57 (20)	47 (23)
M3L2	4 (30)	-6 (23)	-18 (27)

transition onset with respect to the acoustic syllable onset (M1A1 ranging from 60 to 63%rel): these means are statistically comparable as it is shown by a non-significant result of ANOVA ($F < 1$). For the three subjects, the hand transition ends after the syllable onset (A1M2 from 6 to 18%rel), more precisely in the first part of the consonant. A one-way ANOVA shows that A1M2 intervals are somehow different ($F(2, 474) = 14.9$, $p < .0001$). Post-hoc comparisons (Scheffé) showed that the behavior of subject 3 differs from that of the other two ($p < .01$); with respect to acoustic consonant onset the hand target position is reached later for this subject. For the three subjects, the hand position is on average attained well before the vowel lip target (M2L2 in the range of 47 to 62%rel). Statistically, the durations appear to be different (ANOVA, $F(2, 474) = 18.7$, $p < .0001$). The post-hoc tests again show that it is the behavior of subject 3 which differs from the others ($p < .01$); the hand target position anticipation over the lip target appears to be less important. Finally, the three subjects seem to demonstrate more variability concerning the moment the hand leaves the position toward the next position: the ANOVA applied shows that the M3L2 durations are different ($F(2, 474) = 24.8$, $p < .0001$). Again, the post-hoc multiple comparisons show that subject 3 differs from the others ($p < .01$); for this subject, with respect to the vocalic lip target, the hand leaves the position for the following one later than for the other two subjects. Thus the movement onset of the hand seems not to be related to the vocalic target on the lips: rather the hand begins the transition during the acoustic vowel realization. The differences found for subject 3 reveal that this cuer tends to make longer hand transitions and this way, the temporal pattern of hand and speech coordination is shifted back.

4   Discussion and Conclusion

This work describes the investigation of French Cued Speech production in three different subjects certified in manual cueing. We focused on the temporal organization of the hand gestures with respect to lip gestures and acoustic events in the CV syllabic domain. The extracted temporal intervals were considered in milliseconds as well as in percentages relative to the syllable duration. This makes it possible to consider the different syllable durations obtained between the different utterances and across subjects, allowing us to deal with intra- and inter-speaker variability during production of cued syllables.
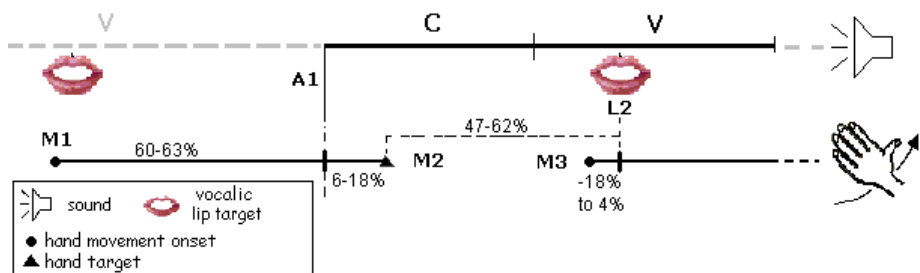
Fig. 3. General temporal pattern of FCS production from results of the three FCS speakers. The temporal events for sound, lip gesture and hand transition are indicated with the range of mean values of intervals (in percentage of the CV syllable duration).

The three subjects were recorded with a wide corpus involving both hand transitions and finger gestures. The investigation focused on hand transitions. Concerning speech rhythm, the mean value of 4 Hz obtained confirms the slowing down of speech during Cued Speech, which was already observed by [7] who indicates a value of 100 wpm, i.e. a range between 3 to 5 Hz for the syllabic rhythm. Concerning the organization of FCS production, each subject reveals a pattern of coordination very similar to the pattern previously described for a single subject [8], with comparable values for each interval. At the statistical level, subject 3 appears to slightly differ from the other two subjects. It seems that this talker has slower transitional hand gestures: this difference could be explained by the level of FCS exercise. Indeed, subject 3 practices FCS for middle school/ junior high school (*collège*) students, whereas subject 1 and subject 2 practice FCS at the high school (*lycée*) level, where the difficulty level and scholarly rhythm are incontestably higher. Despite these differences, the three subjects demonstrated a very similar temporal pattern of coordination.

So the general pattern of hand and speech coordination proposed for a cued CV syllable is the following (also illustrated in Fig. 3):

1. the hand begins its movement before acoustic onset of the syllable;
2. the hand attains its position in the first part of the consonant;
3. the position hence is reached before the vocalic lip target;
4. and finally the hand leaves the position during the vowel.

Proposition 1 appears to be the more consistent across the subjects, with very similar values for this interval (across the subjects, from 60 to 63%rel of the syllable duration). It ensues that a temporal constraint should be that the hand transition onset occurs prior to acoustic onset of the syllable so that the duration between these two events represents around 60 percent of the whole cued CV syllable duration. This was the general behavior observed for each subject concerning the hand movement onset. Proposition 2 is always validated by the three different subjects. The interval duration obtained for each subject was not exactly the same but the constraint here should be that the hand

must point to the spatial target position during the first part of the consonant. There is here a temporal "rendez-vous" between the hand position and the consonant onset. Since the cued position is attained during the consonant, proposition 3 that the hand anticipates the vocalic lip target is always validated, even if the interval between the manual target and the labial target can differ according to the subject. So it appears that the anticipatory behavior of the hand is a general rule of FCS production. The hand position delivers the manual vocalic information before the lip shape delivers the labial vocalic information. Finally, proposition 4 that the hand begins the transition during the acoustic production of the vowel is also validated by all the subjects.

We can conclude that the anticipatory relationship that was found in the single subject study [8] is not idiosyncratic to this individual, but is found consistently on other proficient cueing speakers. It should be noticed that a manual advance was also suggested by [7] in their automatic cueing system ("... human cuers often begin to form cues well before producing audible sound", p. 491). More generally, the manual anticipation is also found in contexts different from Cued Speech: for example for co-verbal iconic gestures related to speech, it appears that the hand begins the gesture more than 200 ms before speech [14]. According to [15], the gesture onset never occurs after the speech onset. So it seems that this anticipatory behavior of the hand is a general feature of manual gesture. For co-speech gestures, this coordination can reflect the common origin of gestures and speech which can take place at different levels of computational stages of speech production depending on the type of gestures considered [14], [15]. However Cued Speech represents a unique system that very tightly links the hand to speech. The common function is not at the semantic level, like for the most part of co-speech gestures, but acts at the phonological level, since the Cued Speech code (hand position and handshape) is determined by speech phonemes by definition. We have found that this "artificial" manual system is completely anchored to the natural speech, with the hand position target and the consonant onset clearly phase-locked. According to us, this coordination can result from an optimal hand-speech control strategy linked to the types of neural motor control (local and postural controls; [16]) of consonants and vowels in Cued Speech and visible speech (see [8] for a detailed discussion). In this view, the vocalic manual contact control and the consonantal contact control of visible speech, which are compatible types of motor control, are synchronized. Obviously this hypothesis needs further investigations particularly in the field of neural control of Cued Speech.

A first study on cued speech perception, using a gating paradigm, allows us to propose that this specific temporal organization is retrieved and used by deaf perceivers decoding FCS ([17]). Preliminary results showed that the deaf perceivers did exploit the manual anticipation: perception of the hand gives first a subgroup of possibilities for the phonemes pronounced; the lips then give the unique solution. It therefore seems that the organization of FCS in production is used for the perception.

Acknowledgments

References

1. Cornett, R.O.: Cued Speech. American Annals of the Deaf **112** (1967) 3–13
2. Cornett R.O.: Adapting Cued Speech to additional languages. Cued Speech Journal **5** (1994) 19–29
3. Nicholls, G., Ling, D.: Cued Speech and the reception of spoken language. Journal of Speech and Hearing Research **25** (1982) 262–269
4. Uchanski, R.M., Delhorne, L.A., Dix, A.K., Braida, L.D., Reed, C.M., Durlach, N.I.: Automatic speech recognition to aid the hearing impaired: Prospects for the automatic generation of cued speech. Journal of Rehabilitation Research and Development **31** (1) (1994) 20–41
5. Leybaert, J., Alegria, J.: The Role of Cued Speech in Language Development of Deaf Chil-dren. In: Marschark, M., Spencer, P. E. (eds.): Oxford Handbook of Deaf Studies, Language, and Education. Oxford University Press (2003) 261–274
6. Bratakos, M.S., Duchnowski, P., Braida, L.D.: Toward the automatic generation of Cued Speech. Cued Speech Journal **6** (1998) 1–37
7. Duchnowski, P., Lum, D., Krause, J., Sexton, M., Bratakos, M., Braida, L.D.: Development of speechreading supplements based on automatic speech recognition. IEEE Transactions on Biomedical Engineering **47** (4) (2000) 487–496
8. Attina, V., Beautemps, D., Cathiard, M.-A., Odisio, M.: A pilot study of temporal organization in Cued Speech production of French syllables: Rules for a Cued Speech synthesizer. Speech Communication **44** (2004) 197–214
9. Lallouache, M.T.: Un poste visage-Parole couleur. Acquisition et traitement automatique des contours des lèvres. Doctoral thesis, INP Grenoble (1991)
10. Abry, C., Boë, L.-J.: "Laws" for lips. Speech Communication **5** (1986) 97–104
11. Badin, P., Motoki, K., Miki, N., Ritterhaus, D., Lallouache, M.-T.: Some geometric and acoustic properties of the lip horn. Journal of Acoustical Society of Japan (E) **15** (4) (1994) 243–253
12. Schmidt, R.A.: Motor control and learning: A behavioural emphasis. Champaign, IL: Human Kinetics (1988)
13. Perkell, J.S., Matthies, M.L.: Temporal measures of anticipatory labial coarticulation for the vowel /u/: Within- and cross-subject variability. The Journal of Acoustical Society of America **91** (5) (1992) 2911–2925
14. Butterworth, B.L., Hadar, U.: Gesture, speech, and computational stages: A reply to McNeill. Psychological Review **96** (1) (1989) 168–174
15. Morrel-Samuels, P., Krauss, R.M.: Word familiarity predicts temporal asynchrony of hand gestures and speech. Journal of Experimental Psychology: Learning, Memory and Cognition **18** (3) (1992) 615–622

16. Abry, C., Stefanuto, M., Vilain, A., Laboissière R.: What can the utterance "tan, tan" of Broca's patient Leborgne tell us about the hypothesis of an emergent "babble-syllable" downloaded by SMA? In: Durand, J., Laks, B. (eds.): Phonetics, Phonology and Cognition. Oxford University Press (2002) 226–243
17. Cathiard, M.-A., Attina, V., Abry, C., Beautemps, D.: La Langue française Parlée Complétée. (LPC): sa coproduction avec la parole et l'organisation temporelle de sa perception. Revue PArole, num spécial 29–30–31, Handicap langagier et recherches cognitives: apports mutuels, (2004–to appear)

Appendix

Table 3. Complete stimulus materials

mamapija	mimipaji	mumupaju	mømøpajø	memepaje
mamapuja	mimipuji	mumupiju	mømøpijø	memepije
mamapøja	mimipøji	mumupøju	mømøpujø	memepuje
mamapeja	mimipeji	mumupeju	mømøpejø	memepøje
mamajipa	mimijapi	mumujapu	mømøjapø	memejape
mamajupa	mimijupi	mumujipu	mømøjipø	memejipe
mamajøpa	mimijøpi	mumujøpu	mømøjupø	memejupe
mamajepa	mimijepi	mumujepu	mømøjepø	memejøpe
mamasila	mimisali	mumusalu	mømøsalø	memesale
mamasula	mimisuli	mumusilu	mømøsilø	memesile
mamasøla	mimisøli	mumusølu	mømøsulø	memesule
mamasela	mimiseli	mumuselu	mømøselø	memesøle
mamalisa	mimilasi	mumulasu	mømølasø	memelase
mamalusa	mimilusi	mumulisu	mømølisø	memelise
mamaløsa	mimiløsi	mumuløsu	mømølusø	memeluse
mamalesa	mimilesa	mumulesu	mømølesø	memeløse
mamaviga	mimivagi	mumuvagu	mømøvagø	memevage
mamavuga	mimivugi	mumuvigu	mømøvigø	memevige
mamavøga	mimivøgi	mumuvøgu	mømøvugø	memevuge
mamavega	mimivegi	mumuvegu	mømøvegø	memevøge
mamagiva	mimigavi	mumugavu	mømøgavø	memegave
mamaguva	mimiguvi	mumugivu	mømøgivø	memegive
mamagøva	mimigøvi	mumugøvu	mømøguvø	memeguve
mamageva	mimigevi	mumugevu	mømøgevø	memegøve
mamabima	mimibami	mumubamu	mømøbamø	memebame
mamabuma	mimibumi	mumubimu	mømøbimø	memebime
mamabøma	mimibømi	mumubømu	mømøbumø	memebume
mamabema	mimibemi	mumubemu	mømøbemø	memebøme
babamiba	bibimabi	bubumabu	bøbømabø	bebemabe
babamuba	bibimubi	bubumibu	bøbømibø	bebemibe
babamøba	bibimøbi	bubumøbu	bøbømubø	bebemube
babameba	bibimebi	bubumebu	bøbømebø	bebemøbe

Using Signing Space as a Representation for Sign Language Processing

Boris Lenseigne and Patrice Dalle

IRIT, Université Paul Sabatier, 118, route de Narbonne,
31062 Toulouse cedex 9, France

Abstract. Sign language processing is often performed by processing each individual sign. Such an approach relies on an exhaustive description of the signs and does not take in account the spatial structure of the sentence. In this paper, we will present a general model of sign language sentences that uses the construction of the signing space as a representation of both the meaning and the realisation of the sentence. We will propose a computational model of this construction and explain how it can be attached to a sign language grammar model to help both analysis and generation of sign language utterances.

1 Introduction

Sign languages, such as the French sign language, use gestures instead of sounds to convey a meaning. They are deaf peoples' natural languages. Unlike oral languages, sign languages are characterized by a great multilinearity due to the fact that the signer can simultaneously use several body parts to communicate: hand configuration, localisation and motion, facial expression, body motion,. . .

Most of the time, one considers two levels of language: standard utterances that only use standard signs, the ones that can be found in dictionaries, and iconic utterances, so-called "classifier predicates", where most of the meaning relies on iconic structures. Iconic structures are widely used in spontaneous sign language so that they need to be taken in account in automatic sign language processing systems.

Works in French Sign Language (FSL) linguistics [4][3] have shown that, in both standard and iconic utterances, the meaning of a sign language production could be accessed by considering the construction of the signing space. The signing space is the space surrounding the signer and where the signs are produced. During this production, the signer will use that space to position the entities that are evoked in the sentence and to materialize their semanting relationships, so that the resulting construction can be considered as a representation of the meaning of the discourse.

In this paper, we propose a computational representation of this organisation, and describe how this representation can be used to help both automatic interpretation and generation of sign language.

2   Previous Work

Most of existing works on the description of sign language focus on describing, as precisely as possible, the gestures that are used to produce the signs. Linguistic sign description systems such as W.S. Stokoe's [5] or, more recently, HamNoSys [20] led to descriptions based on discrete sets of parameters and values to describe signs (hand configurations, elementary motions,...).

In the field of sign language automatic processing, those description systems have been used as templates to define the primitives that have to be characterized for sign recognition. For instance, C. Vogler and D. Metaxas [6][7] use the Liddel&Johnson description system [21] and R.H. Liang, M. Ouhyoung [18] the Stokoe notation, to define the primitives to identify. In the field of sign language utterrances generation, the VisiCast project is based on SigML which is an XML implementation of the HamNoSys notation system [10][11].

Many other works, especially in the field of vision-based sign recognition, use specific feature vectors depending on the data available for recognition, as in [1] or [17]. Only a few one take in account the spatial structure of the utterance. In the field of automatic translation, H. Sagawa et al. [19] propose a vision-based sign recognition system that handles directional verbs and A. Braffort's sign language traduction system, Argo [2], is able to translate iconic sentences that have a fixed structure, by using a single dataglove. The sign language generation system by S. Gibet and T. Lebourque [22] allows the generation of spatially arranged sentences by including the notion of targets to specify gestures which enables the use of explicit designations and directional verbs. Finally, M. Huenerfauth [13] proposes a classifier predicate generation system based on a virtual reality system that is used to specify the relative locations of the entities evoked in the predicate.

But, for the moment, none of these works led to a global model of sign language utterances spatial structure.

3   A Computational Signing Space Representation

The initial goal of this modelisation was to provide a internal representation of a sign language sentence for vision-based sign language analysis purposes. To avoid the risk of an abusive simplification implied by an incomplete description, it focuses on representing a subset of the sign language. This subset concerns sentences produced in the context of a timetable, that means sentences that brings on play persons, places, dates and actions.

3.1   What to Represent in That Model ?

The construction of the signing space is mainly useful to represent the relationships between the entities evoked in the discourse, which can be done without knowing the exact kind of those entities. From this point of view, the signing space representation does not have to represent exactly every notion evoked in
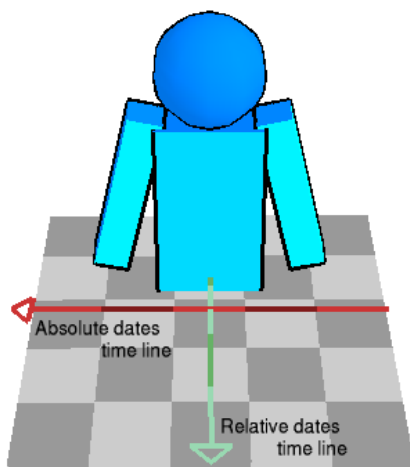
Fig. 1. Symbolic view of the signing space showing the time lines where the dates may be located

the signed production. It is limited to those that may be involved in some relationship, so that the entities can be distinguished by the relationships that may link them.

From a cognitive point of view [23][14][9], there are only a few possible relationships that can be evoked in a sentence :

- *Temporal relationships* that can be either absolute or relative to the current time of the production.
- *Spatial locations* between two entities.
- *Actions* that can link two or more entities.

In the FSL, entities are evoked through signs and located in the signing space so that their relative position will correspond with the spatial relationships between those entities in the real world. Temporal relationships are evoked through entities that are located on "time lines" (fig. 1). Finally binary actions are evoked through directional verbs and more complex ones by grammatical structures called "transfers" in [3].

Different kinds of entities depend on the kind of the relationships in which they may be involved :

- *dates* can be involved in temporal relationships ;
- *places* in spatial relationships ;
- *animateds* can perform an action or be located in relation to another ;
- finally *actions* can be referenced as a moment in time or as a protagonist of an action.

The specificities of the French sign language grammar require to consider some additional kind of entities: one needs to make a distinction between entities that whenever involved in a complex action are evoked by the signer taking

Table 1. Different kinds of entities that may be evoked in a signed sentence and relationships that can exist between them

Entity	Potential relationships			
	Relative temporal location	Absolute temporal location	Spatial location	Action
Date	×	×		
Place			×	
Animate			×	×
Person			×	×
Action		×		×
Object			×	×
Implicit			×	×

their role($persons^1$) and the entities that cannot be evoked that way (*objects*). Finally, due to the temporal ordering of the signs, one needs to take in account the case of actions that are evoked before one of their protagonists. This entity has an *implicit Type*.

Table 1 gives an overview of the different kinds of entities that can be evoked depending on the relationships that may link them.

3.2   General Structure of the Model

The symbolic representation of the signing space consists in a cube surrounding the signer, regularly divided into *Site*(s)[2]. Each location may contain a single *Entity*, each *Entity* having a *Referent*. A *Referent* is a semantic notion that can be found in the discourse. Once it has been placed in the signing space, it becomes an *Entity* and has a role in the sentence. So that, building a representation of a sign language sentence consists in creating a set of *Entities* in the *SigningSpace*. The meaning contained in this signing space construction is represented in terms of *Entities*(s) whose *Referent*(s) can have successively different *function*(s) during the construction of the sentence (*locative, agent*,...). A set of rules maintains the consistency of the representation by verifying that enough and coherent information has been provided when one needs to create a new entity in the signing space. The figure (fig. 2) describes the global architecture of the model in the UML notation standard.

3.3   Creating a New Entity in the Signing Space

Every time a new entity is created in the signing space, specific mechanisms are used to ensure the consistency of the new signing space instanciation. Those mechanisms depend on the type of that entity.

---

[1] In French sign language, persons are not necessarily humans, they can be assimilated to animals or even objects of the real world in humorous stories for example.

[2] Terms written using a *slanted* font are elements of the model.
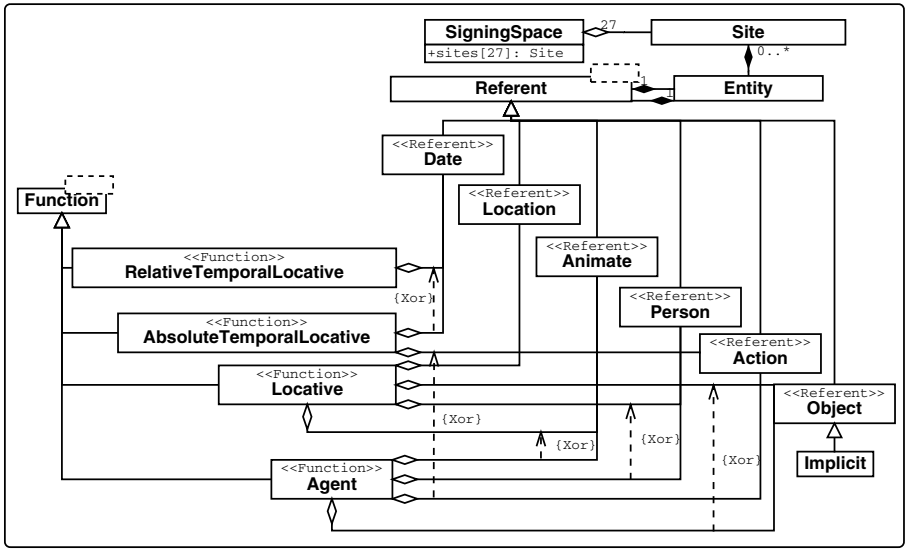
Fig. 2. UML class diagram of the semantic representation of the *SigningSpace*. The *SigningSpace* is regularity divided into *Sites*. Each *Site* can contain a single *Entity* whose *Referent* can have several *Function*(s) during the realisation of the sequence.

Creating a Generic Entity. Generic entities, that are neither *Date*(s) nor *Action*(s), can have two *Function*(s) : *Locative* or *Agent*. Their default function is *Locative*. When such an entity is created in a given *Site*, a new *Referent* of the given *Type* is created. In the case of an automatic analysis system that doesn't take in account the lexicon, it not possible to determine the exact *Type*
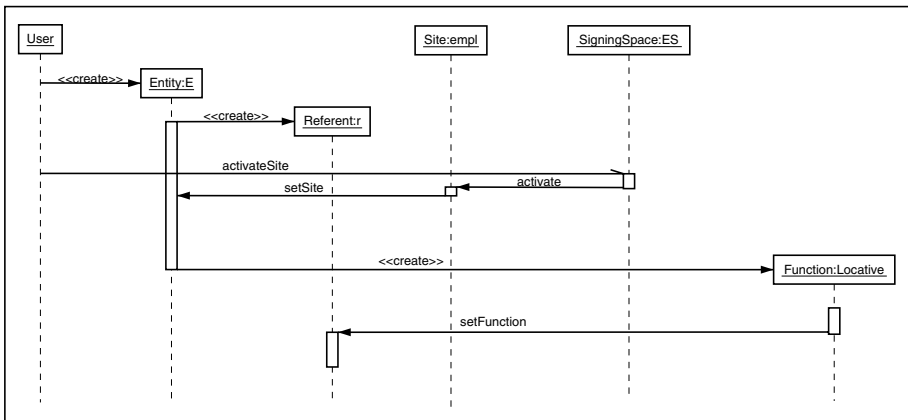
Fig. 3. UML sequence diagram describing the modications of the signing space resulting of the creation of a generic *Entity* in that space
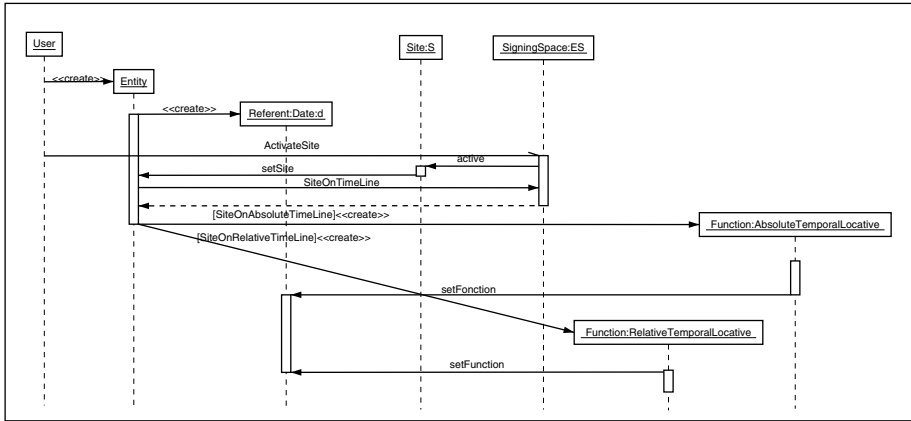
Fig. 4. UML sequence diagrams describing the modications of the signing space resulting of the creation of a new *Date* in that space

of the entity so that it remains *unknown* and can potentially be implied in every kind of relationship. The exact *Type* of such an entity will be later changed depending on its successive *Function*(s) during the production of the utterance. The mecanisms that lead to the creation of such an entity are described as an UML sequence diagram in figure 3.

Creating a *Date*. The modifications of the *SigningSpace* in the case of the creation of a *Date* are quite the same as those used to create a generic *Entity*. Their default *Function*, which can be either an *Absolute* or a *Relative* temporal locative, depends on its location on one of the time lines (fig. 1). The details of those mechanisms are given in figure 4.

Creating an *Action*. *Action*(s) don't have their own *Location*. They link several entities depending on the *Arity* of the *Action*. Those entities are *Protago-*
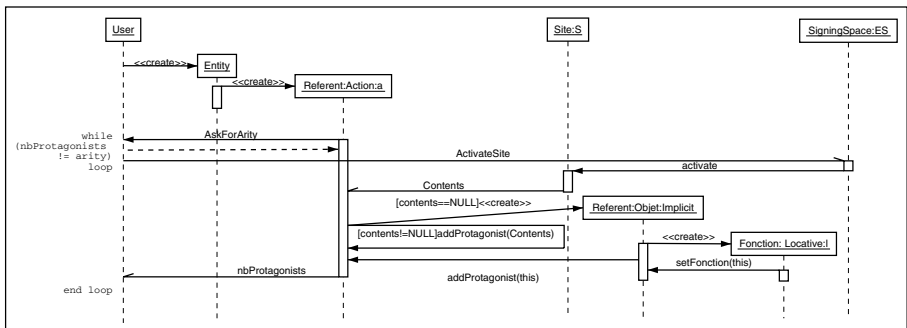
Fig. 5. UML sequence diagrams describing the modications of the signing space resulting of the creation of a new *Action* in that space

nist(s) of the *Action* and their new *Function* is *Agent*. *Protagonists* are defined thru the *Site*(s) that are activated when the signer evokes the *Action*. In the case of a *Site* being empty, an *Implicit Entity* is created in this *Site* (fig. 5).

3.4   An Example of the Construction of the Signing Space

As an illustration of the use of that model, we will now describe the construction of a *SigningSpace* corresponding to an utterance that concerns a question on
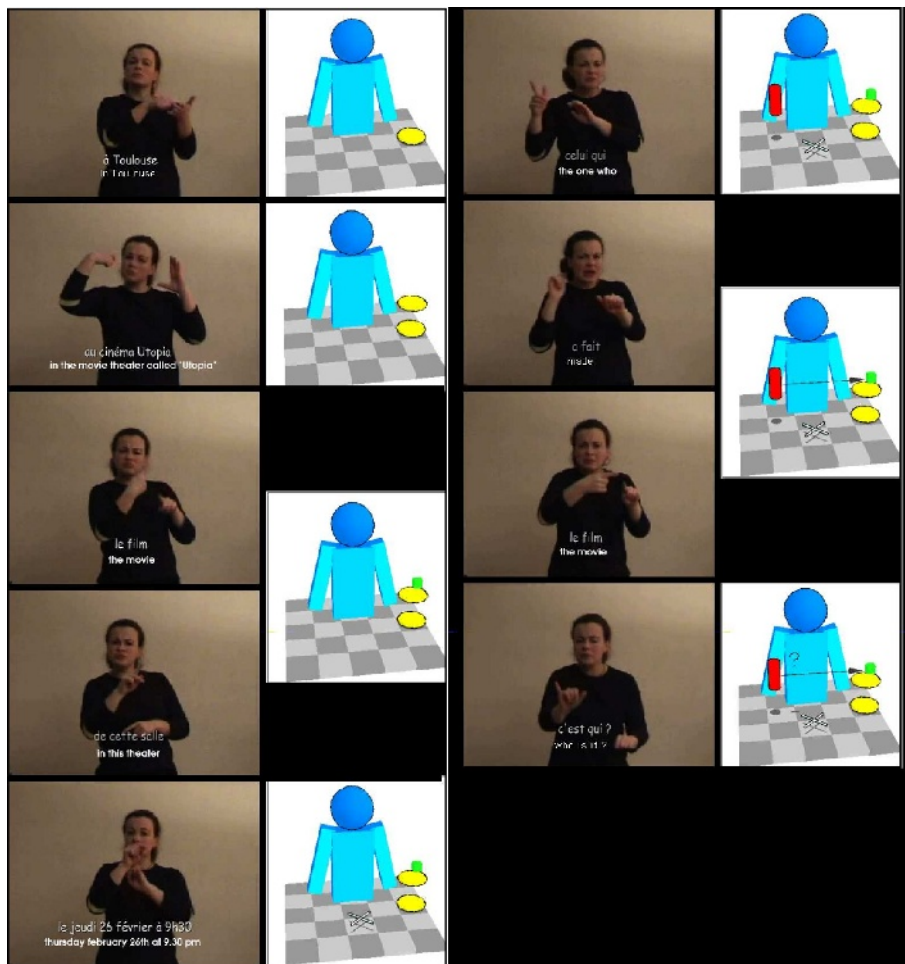
Fig. 6. An example of the construction of the signing space during the realisation of a sign language sentence built with the interactive video sequences transcription tool 4. The sign-to-word translation of that sentence is : *"In Toulouse (1) - In the movie theatre called Utopia (2) - The movie (3) - In this theater (4) - Thursday february the 26th at $9:30$ pm (5) - the one who (6) - made (7) - That movie (8) - Who is it ? (9).*

Table 2. Geometric primitives that are used to represent different kinds of entities in the 3*D* representation of the *SigningSpace*

Type of the entity	Geometric primitive
Date	white horizontal cross
Place	yellow horizontal disc
Personne	red vertical cylinder
Action	arrow
Object	green cube

a cinema program : *"Who is the director of the movie that plays at* 9.30 *pm on thursday february the* 26th *in Toulouse at the Utopia"*. The sign-to-word translation of that sentence is : *"In Toulouse (1) - In the movie theatre called Utopia (2) - The movie (3) - In this theater (4) - Thursday february the* 26th *at* 9 : 30 *pm (5) - the one who (6) - made (7) - That movie (8) - Who is it ? (9)*. The successive configurations of the *SigningSpace* are represented by 3*D* scenes in figure 6. In this *SigningSpace* representation, each kind of *Entity* corresponds to a geometric primitive as detailed in table 2. This transcription was made using an interactive application that allows to manually build the signing space and that implements our model. During the production of the utterance, entities are successively created in the signing space by the mean of signs or specific grammatical structures in the following order :

1. The first *Entity* to be created is a *Place* that is located on the left of the *SigningSpace* and that corresponds to the city of Toulouse.
2. A second *place* is created in the same *Site* of the *SigningSpace* this means that the movie theater is located in Toulouse.
3. The movie that plays in that theater is prepresented by an *Object* that is located in the same *Site* as the movie theater, thus meaning that the movie plays in that movie theater.
4. The *Date* is located in front of the signer in a *Site* that is located on the time line.
5. An *Entity* whose *Type* is *Person* is created and located on the right of the *SigningSpace*.
6. The person is linked to the movie by the mean of an *Action* so that both the person and the movie become *Protagonist*(s) of that *Action*. Their *Function* is changed into *Agent*.
7. Finally, the question that concerns the person is evoked in the same *Site* as the person. Note that entities whose *Type* is *question* are specific to the interactive implementation of our model.

4   Using the Model

The representation of the signing space can be linked to the meaning of the discourse by giving access to the relationships between entities that were evoked and

referenced. On the other hand, the iconicity theory by C. Cuxac [4][3] provides a description of the grammar of the sign language in terms of gesture sequences that leads to creating a new entity in the signing space so that it permits to link this representation to the gestures that where used to create the current signing space instantiation [15][16]. Such a predictive model can be used for both analysis and generation of sign language sentences.

Using the Signing Space for Sign Language Analysis. Using that model for sign language analysis leads to two classes of tools : interactive tools intended for linguists to evaluate the model and automatic analysis tools that can be used in many fields of application (linguistic analysis, automatic interpretation,. . .).

At present time, an interactive tool has been developed in order to represent the construction of the signing space during the production of the utterance. This tool consists in a transcription software that allows to synchroneously link the different steps of the construction of the signing space and the video sequence that is transcripted. This application was designed to evaluate the model on several kinds of utterances and to determine how this model can be considered as a generic representation of sign language utterances.

In the field of automatic analysis, due to the fact that it is not possible, using a single camera, to build an exhaustive description of the gestures that are used, for automatic vision-based sign language analysis, the model of the signing space is used as a general representation of the structure of the sentence that allows simultaneously to access the meaning of the discourse. The grammar of the sign language that can be attached to that construction allows the use of a prediction/verification approach [16][8]: being given an hypothesis on the meaning of the discourse in terms of a signing space modification, it is possible to infer the gestures that where used to create the new entity in the signing space. Analysing the utterance is then reduced to verify whenever the data corroborates this prediction or not. Such an analysis can be performed without taking in account the lexicon, so that the gestures' descriptions that can be used need to be less precise that the ones required for exhaustive sign recognition. This makes the analysis of low resolution images possible.

However, in a reduced context, the spatial structure of the sentence may be a interesting guideline to identify the signs as it can be done by only considering discriminative aspects of the signs. For instance, by requesting a database concerning airline travels, *places* will be evoked by the name of the towns that can be identified by only considering hand positions.

The three different elements of such automatic tool (signing space representation, grammatical model, low level image processing) have been evaluated separately. It has been shown that in a reduced context, the prediction/verification approach that is described above was relevant and allowed to use simple $2D$ image processing operators instead of complex gesture reconstruction alorithms to performs the identification of the different kinds of entities that where used in the utterance.

Using the Model for Sign Language Generation. For sign language synthesis, signing space modelling may be used to describe the general structure of the sentence to generate : the sentence is described as a temporally ordered sequence of the entities' creation. So that it is possible to attach to each entity's creation a preliminary description of the underlying gestures that will be used to constrain the generation of the signs properly speaking. This approach provides an easy-to-use way to describe the sentence to generate and will lead to produce spatially organized sentences that are much closer to natural sign language productions than simple coarticulated sign sequences.

The use of the model for sign language generation purposes has been studied in several fields of applications, but the existing elements of the sign language model have not been included in any sign language generation system for now.

5   Conclusion

By looking for a sign language representation that could be used for sign language image analysis, we proposed a general model of the structure of sign language sentences that takes in account the spatial organisation of those utterances. As this representation can be attached to the gestures that where used to produce the sentence, it constitutes a generic model of the sign language grammar that does not need a fine gesture description. The predictive nature of this model makes it useful for both analysis and generation of sign language.

Moreover one of the main interests of our signing space representation is the possibility to use it, eventually interactively, as a transcription system for sign language sentences. This aspect suggests a new approach to study a written form of the sign language as well as for linguistic studies on the sign language grammar. The integration of the sign language model in a sign language analysis system requires its formalisation and will enable the linguists to collate the linguistic assumptions on this grammar to its application through the interpretation of the sequence.

Finally, gesture descriptions that are used in this model rely on functionnal terms such as "pointing out a place" or "produce a sign in a given place" rather than perceptual terms [12], which point out the interest of this kind of approach for gesture interpretation.

References

1. T. Starner A. Pentland. Real-time american sign language recognition from video using hidden markov models. Technical Report TR-375, M.I.T Media Laboratory Perceptual Computing Section, 1995.
2. A. Braffort. Argo : An architecture for sign language recognition and interpretation. In P. Harling and al., editors, *Progress in Gestual Interaction*, pages 17–30. Springer, 1996.

3. C. Cuxac. French sign language: proposition of a structural explanation by iconicity. In Springer: Berlin, editor, *Lecture Notes in Artificial Intelligence : Procs 3rd Gesture Workshop'99 on Gesture and Sign-Language in Human-Computer Interaction*, pages 165–184, Gif-sur-Yvette, France, march 17-19 1999. A. Braffort, R. Gherbi , S. Gibet , J. Richardson, D. Teil.

4. C. Cuxac. *La langue des Signes française. Les voies de l'iconicité.* ISBN 2-7080-0952-4. Faits de langue, Ophrys, Paris, 2000.

5. W. Stokoe D. Casterline, C. Croneberg. *A dictionnary of american sign langage.* Gallaudet College Press, 1965.

6. C. Vogler D. Metaxas. Toward scalability in asl recognition: Breaking down sign into phonemes. *Gesture Workshop'99, Gif-sur-Yvette, France*, mars 1999.

7. C. Vogler D. Metaxas. Handshapes and movements: Multiple-channel asl recognition. In G Volpe A. Camurri, editor, *Proceedings of the Gesture Workshop'03, Lecture Notes in Artificial Intelligence*, volume 2915 of *Lecture Notes in Computer Science*, pages 247–258, Genova, Italy, April 15-17 2003. Springer.

8. P. Dalle and B. Lenseigne. Vision-based sign language processing using a predictive approach ans linguistic knowledge. In *IAPR conference on Machine vision and application (MVA 2005)*, pages 510–513, Tsukuba Science City, May 2005.

9. J.P. Desclés. La prédication opérée par les langues. In *Langages*, pages 83–97. Larousse, Paris, 1991.

10. J.A. Bangham et al. An overview of ViSiCAST. In *IEEE Seminar on Speech and language processing for disabled and elderly people*, London, April 2000.

11. R. Elliot et al. An overview of the SiGML and SiGMLsigning software system. In *Workshop on Representation and Processing of Sign Language, 4th Internationnal Conference on Language Ressources and Evaluation (LREC 2004)*, pages 98–103, Lisbone Portugal, 30 May 2004.

12. B. Lenseigne F. Gianni, P. Dalle. A new gesture representation for sign language analysis. In *Workshop on Representation and Processing of Sign Language, 4th Internationnal Conference on Language Ressources and Evaluation (LREC 2004)*, pages 85–90, Lisbon Portugal, 30 may 2004.

13. M. Huenerfauth. Spatial representation of classifier predicates for machine translation into american sign language. In *Workshop on Representation and Processing of Sign Language, 4th Internationnal Conference on Language Ressources and Evaluation (LREC 2004)*, pages 24–31, Lisbon Portugal, 30 May 2004.

14. R. Langacker. *Foundations of cognitive grammar: Theoretical Prerequisites.* Stanford University Press, Stanford, CA, 1987. Vol 1, 1987 (Hardcover), 1999 (Paperback).

15. B. Lenseigne. *Intégration de connaissances linguistiques dans un système de vision. Application à l'étude de la langue des Signes.* PhD thesis, Université Paul Sabatier, Toulouse, décembre 2004.

16. B. Lenseigne and P. Dalle. A model for sign language grammar. In *2nd Language and technology Conference*, Poznan Poland, April 21-23 2005.

17. R. N. Whyte M. G. Somers. Hand posture matching for irish sign language interpretation. In *ACM Proceedings of the 1st international symposium on Information and communication technologies*, pages 439–444, Dublin Ireland, September 24-26 2003. Trinity College Dublin.

18. R.H. Liang M. Ouhyoung. A real-time continuous gesture recognition system for sign language. In 3rd *International conference on automatic face and gesture recognition*, pages 558–565, Nara, Japan, 1998.

19. H. Sagawa M. Takeuchi, M. Ohki. Description and recognition methods for sign language based on gesture components. In *in Proceedings of IUI*, orlando, Florida, 1997.
20. S. Prillwitz and al. *HamNoSys. Version 2.0; Hamburg Notation System for Sign Languages. An introductory guide.* Hamburg : Signum, 1989.
21. S. Liddell R. E. Johnson. American sign language: The phonological base. *Sign Language Studies*, 64:195–227, 1989.
22. S. Gibet T. Lebourque. High level specification and control of communication gestures : the GESSYCA system. In *Computer Animation'99*, Geneva, 26-28 May 1999. Computer Graphics Society (CGS) and the IEEE Computer Society.
23. R. Thom. *Stabilité structurelle et morphogénèse.* Édisciences, Paris, 2e edition, 1972-1977.

Spatialised Semantic Relations in French Sign Language: Toward a Computational Modelling

Annelies Braffort and Fanch Lejeune

LIMSI/CNRS, Orsay, France
annelies.braffort@limsi.fr
http://www.limsi.fr/Individu/braffort/ATGeste/

Abstract. This paper belongs to computational linguistics, with reference to French Sign Language (FSL) which is used within French deaf community. Our proposed modelling is intended: to provide computational modelling of a se-mantico-cognitive formalisation of the FSL linguistic structure, and to allow its implementation and integration in applications dedicated to FSL, such as analysis in image processing, automatic recognition, generation of writing forms or realisation by signing avatars.

1 Introduction

French Sign Language (FSL) is the language practised by the French deaf community. Like all Sign Languages (SL) around the world, FSL intensively uses the signing space located in front of the signer in order to describe entities, express semantic relations, and conjugate directional verbs. Several "articulators" are simultaneously used to elaborate and structure discourse (hands, arms, gaze, mimic, chest), each of them having one or several specific linguistic roles, and possibly interacting with other ones.

Due to its visuo-gestural channel specificity, FSL allows to "tell", but also to "show while telling" [1]. This capacity is typically exploited in semantic relations between entities, allowing the signer to explicitly show the relation in the signing space. A simple example of such spatialised semantic relations is the spatial relation (i.e. "in", "on", "near"…) between two entities. The relation is not express directly between the entities, but specific configurations performed at selected locations in the signing space [1]. These configurations are not chosen without reason, and utterance structure is not arbitrary.

This paper describes a formal representation of such relations, as well as a computational model. The proposed modelling has two aims: to provide a computational modelling of a semantico-cognitive formalisation of the linguistic structure of FSL, and to allow its implementation and integration in applications dedicated to FSL, such as analysis in image processing, automatic recognition, generation of writing forms or realisation by signing avatars.

The next section relates this work to similar studies. Our formalism is introduced within Section 3 and the computational model in Section 4. The last section offers our conclusion and prospects.

2   Related Studies

If linguistic studies devoted to particular aspects of sign languages such as iconicity or space structure have been published for various SL, like FLS [1][2][3][4][5], or American Sign Language (ASL) [6][7][8], research in computational linguistics dedicated to these aspects is relatively recent and is much less developed.

Most of the studies on computational aspects are carried out in the context of a given application, such as automatic generation of SL utterances performed by a signing avatar. In several studies, the proposed modelling of SL structures integrates syntactic representations and sometimes even semantic representations, more or less specific to SL. The following examples concern the translation of text into SL utterances.

In the TGT project [9], there is no specific representation of the structures of SL and the signs order is almost the same as word order in the input sentence. This kind of approach cannot be applied to spatialised sentences. In the Team [10] and the ASL workbench [11] projects, some representations specific to SL are proposed, allowing for example to represent directional verbs. But there is no mean to represent sentences with spatial locating and iconicity.

Other projects aim specific SL structures to be represented. In the European Visicast [12] and then eSign [13] projects, some representations specific to SL are proposed, including non-manual features such as eye gaze, head and shoulders movements, and precise mimics. But the structure of the signing space is for the moment limited to a set of predefined locations. An original approach [14] explicitly proposes a modelling of the signing space. The system, not yet implemented, should be able to produce spatialised sentences including classifier predicates.

In France, the first studies on computational modelling of LSF were also developed in a specific context. In [15], modelling of signing space was proposed to allow the interpretation of directional verb conjugation in the context of automatic recognition. More recent studies propose more generic and advanced models, even if they are also carried out for a specific use (analyses of video sequences [16] or automatic generation [17], [18]).

Most of the time, the proposed models are limited by the constraints due to the application (capture in recognition and animation control in generation). Our study tries to go one step further in the modelling of specific SL structures, like iconicity, use of space and simultaneity of information independently of any application.

3   Semantico-Cognitive Formalisation

In SL, the gestures are performed in a significant way, in a space represented by a half-sphere located in front of the signer. This space makes it possible to represent the whole set of semantic relations between various entities in a relevant and structured way. These relations can be locative, temporal, or relating the different participants in an action. Via various mechanisms (gaze, pointing, modification of the location of lexical signs...), it is possible to build virtual spatial references (or loci). In general,

the entities are not in relation directly, but via these virtual locations. Thus, we have a spatial structuring of the signing space, by means of various locations that can be associated to the entities by using specific configurations, named "proforms" [1].

Our aim is to model the semantico-cognitive processes underlying this type of utterance elaboration. This modelling is based on the formal model elaborated in [19] and applied for the FSL in [18]. In his thesis, F. Lejeune proposes a set of semantico-cognitive representations specific to the FSL. One of these representations relates to the spatialised semantic relations, like in "the cat is in the car", or even in more abstract utterances like in "Several disciplines, such as linguistics, psychology, sociology, belong to human sciences".

In this paper, we will use the example "the cat is in the car" to introduce the formal and computational models. This utterance is composed of:

- **lexical signs** of FSL designing entities, noted $[sign]_{LSF}$ (figure 1a and 1c),
- **proforms** allowing to spatialise the entities, noted PF(sign) (figure 1b and 1d). The way these proforms are spatialised shows the spatialised relation between the entities (figure 1d),
- and also **gaze**, which is used to "instantiate" a locus in the signing space and which is realised just before performing a proform (just before the images 1b and 1d).

Fig. 1.   a. $[car]_{LSF}$ b. PF(car)   c. $[cat]_{LSF}$ d. PF(car)+PF(cat)

The model is based on the description of salient "situations" from a visual perception point of view. These situations are formalised using semantico-cognitive primitives, which can be properties of the entities, operators and relations between various types of entities.

Semantico-cognitive Type. The entities are typed from a semantico-cognitive point of view. Table 1 lists some of these elementary types.

Table 1. Examples of entity type

Entity types	Examples
Individualisable	cat, car
Location	Ile de Berder
Collective	the people
Massive	water, butter
Activity	work

This type is not static: The situation context can imply a change of type. For example, when an entity *E1* is positioned in the signing space using a proform, in order to be used thereafter as landmark for another entity *E2*, the *E1* type takes the 'Location' value, whatever was its type at the beginning.

In our example, the type of the entities *Car* and *Cat* normally is 'Individualisable'. But in the situation "the cat is in the car", *Car* takes the 'Location' type, because it is used as a landmark for *Cat*.

Operators. Operators are used to specify or change an entity property, or to structure the signing space, by creating and specify particular locations.

Some operators can be used to change an entity type. In our example, one of these operators is used to attribute the 'Location' type to the *Car* entity, whose initial type is 'Individualisable'. This is represented by the expression

LOC(car).

The proforms, not only characterise an entity among several ones, but also provide a particular point of view on this entity, salient in the context of the relation. Thus, the determination of the proforms is not trivial. It will depend in particular on:

- *semantico-cognitive type of the entities*. For example, some configurations, rather "neutrals" such as the configuration 'flat hand, spread fingers', are often used to represent massive entities (for example *water* in "it plunges in the water") or collective entities (for example "a queue of standing persons").
- *context of use*. For example, if one is interested in the interiority of the landmark, the proform used for referring to this entity will be based on a concave configuration, like the 'C' configuration used in the proform in Figure 1 image b.

The 'DET' operator is used to characterise a salient property of an entity. This operation can be seen as an operation of determination. 'DET' is formally characterised by a given proform, or at least a small subset of proforms.

In our example, the cat is specified as an entity located in relation to the car location by the way of a 'X' configuration. In the formalism, this determination is represented by the expression

DET(cat).

The proform retains a salient feature of an entity locating in relation to a landmark. The Car entity is determined as a landmark entity of 'Location' type. In our formalism, this determination is represented by the expression

DET(LOC(car)).

The proform retains a salient feature of an entity as a landmark. In a generic way, the expression
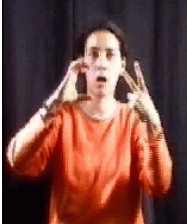
DET(x)

defines a proform which retains a salient feature of an entity x that depends on its type.

It is then possible to specify a little more this determination, by using operators of topological nature. Topology intervenes in linguistics to represent concepts of locating and topological determination. It is defined on abstract locations that transcend the categories of space, time, activities, modalities and notions. For a given location entity, whatever its nature, topological operators are used to specify its interiority (IN), exteriority (EX), border (FR) and globality (FE). These operators can apply only to entities with a 'Location' type that will thus be more specified. In our example, one can specify that we are interested by the inside of the car. That is represented by the expression

IN(DET(LOC(car))).

Table 2. The four linguistic operations in the sentence "the cat is in the car"

Linguistic operation	Gesture unit	Formalisation
1: Lexical sign (that will be used later as *Landmark*)		Car Type: Individualisable

2: Activation of a location loc1 in the signing spaceby means of the gaze, chosen by the enunciator in relation to the statement situation (Sit(S0,T0)) Specification of a relevant property of the landmark and localisation of this proform in the signing space		Car Type: Location
		    Role: landmark
		    Point of view: container
		    Proform: C
		
		Locus1 = REG(loc1) &
		loc1 REP Sit (So, To)
		IN(DET(LOC(car))) REP loc1

3: Lexical sign (that will be used later as *Localised entity*)		Cat Type: Individualisable
4: Activation of a location in the signing space (loc2 = inside loc1), specification of a relevant property of the localised entity and elaboration of the inclusion relation		Cat Role: located Proform: X Locus2 = REG(loc2) & loc2 = IN(loc1) & DET(cat) REP IN(DET(LOC(car)))

In a generic way, the expression

$$IN(DET(x))$$

determines a proform which states that x is considered as a "container".

Relators. Relators are used in particular to express a property concerning two (or more) entities. The main one is the relator REP. The expression 'x REP y' means that the entity x is located in relation to y.

In our example, we express that the *Cat* is located in relation to the inside of the car by using the expression

$$DET(cat) REP IN(DET(LOC(car))).$$

Other relators allow us to express the change from a situation to another, in statements that express for example the displacement of an entity (ex: "the cat jumps in the car"), or the way in which an action is carried out (ex: "I put the cat in the car"). Let us note that in this case, the utterances will differ by the proform used or the movement dynamics (this will not be developed in this article).

Complete example. The use of the formalism to describe sentence structure is detailed for each linguistic operation in the following table (Table 2). We list four main operations, each one containing one or more sub-operations.

The Table 3 represents a scheme that synthesises our example representation. First row is used to describe the two entities (type, attribute), while second row describes the situation: an inclusion relation.

Table 3. Specific LSF scheme for an inclusion relation

1: Cat
Value: Individualisable
Role: located
2: Car
Value: Location
Role: landmark
Point of view: container
<DET(cat) REP (IN(DET(LOC(car))))>

Those schemes are not universal since their organisation is specific to each language. Nevertheless, for simple concepts, one can indeed find categories of generic schemes almost similar between languages. This kind of representations could be used as interlingua representations for application involving a SL and an oral language.

The following scheme (Table 4) gives a generic representation of a situation of spatial relation between two entities.

Table 4. Generic scheme for an inclusion relation

x: Individualisable y: Location
x REP y

The next section describes a first tentative to elaborate a computational modelling of this formalism for FSL.

4   Computational Modelling

The proposed computational modelling is build upon several knowledge bases, relating to the entities, their semantico-cognitive types and their proforms, to the spatialised relations, and the various associated mechanisms.

Its main features are synthetically introduced in a first section using the UML notation, which has the advantage of being able to express in a visual way the various concepts, their structuring, hierarchisation and the relations between these concepts. The *classes* (rectangle) represent the various concepts and the *associations* (lines between the rectangles) represent the relations or dependencies between these concepts.

A second section illustrates how this modelling can be implemented for automatic generation of FSL utterances.

4.1   Model

To represent utterances expressing a spatialised relationship, one must model the spatial structure of the signing space, since entities are spatialised on chosen locations, and then, the properties of entities, proforms, semantico-cognitive type and relations brought into play.

Signing Space. Generally, signing space structure is performed at a lexical level. This consists to provide a list of possible values of the parameters that define visually the lexical signs of the various countries [20] [21].

Other models, elaborated in studies on written forms of the SL [22][23], or on signing avatar animation [24][25], propose a more complex structuration, in order to improve the precision of description.

It still remains to elaborate models that integrate wider knowledge, in particular on the nature of the loci which structure space, their relations, overlapping, degree of accuracy and nature. An original point of view on a functional structure of the signing space is proposed in [16] [26] [27]. It allows the different loci to be specified in a given utterance regarding the functional role of the associated entities in the utterance, such as "person", "action", "date"… For our part, we have focused our study on the way loci are associated in a spatial relation between two entities.

For example, in an utterance which describes an inclusion relationship between two entities, the location of the located entity, referenced using an adequate proform, is determined according to the location of the proform representing the landmark entity, so that the relation is visually not ambiguous. Moreover, it specifies the way in which relation is carried out (completely, partially...). This is here specification mechanisms of iconic nature which are carried out.

In order to integrate such considerations in the model, it seems really necessary to refine the structuration of the signing space, and insert knowledge of higher level. In a

first step, we propose to differentiate the information related on the way the signing space is represented in the application, and the information related on the linguistic role of the loci. Thus, we define these two notion:

- A **Position** refers to a location in the signing space, regarding a reference such as the signer location. It is described by a 3d point or a volume, depending on the way the signing space is segmented in the application.
- A **Locus** represents a point of interest in the signing space, from the linguistic point of view. It is represented by a radius, symbolising a sphere: As an entity can have a volume being more or less extended in the signing space (ex: the sea), the radius allows to precise its range. A Locus can manage operators, such as the 'REG' operator that is used to create or activate an instance of Locus.

A Locus is associated to only one Position. On the contrary, a Position can be associated to several Loci, for example in the case of embedded entities.

Schemes. As explained in section 3, the entities are typed with semantico-cognitive properties. The model integrates knowledge that makes it possible to assign, to modify and to specify the entity type. It also makes it possible to draw up a list of preferential proforms for each entity according to its type and to the nature of the semantic relation. It is illustrated using the simplified class diagram presented in Figure 2.

The first concept is the **Scheme**, which represents the various possible schemes represented in this model, such as the one given in Table 3. It is described by the list of successive operators and relators (LOC, DET, IN, REP...) applied to the operands (the entities). Each scheme is associated to two or three entities according to the relation. In the case of a spatialised relation, each scheme will be associated to two entities.

The **Entity** concept represents the entities handled in the utterance. Each entity is defined by its lexical sign. The sign description (by means of visual or physiologic properties) is not given, because it depends on the applications (generation, analysis, image processing, recognition...).

The concept of **Semantico-Cognitive Type** (or **SCType** in Figure 2) corresponds to the various types quoted previously (in Table 1). For a given context, an association (named **typing** in the diagram) will be established between a given Entity and a given SCType. When an Entity is activated, it receives a preferential SCType. Then, the context can impose a change of type. We use specific operators, such as LOC, to update this association when the entity changes its type.

The **Proform** concept represent the various meaningful points of view associated to entities. Thus, each entity can be associated to several proforms. Conversely, a proform can be associated to several entities. One represents the relation between an Entity, a SCType and a Proform using a ternary association (named **determination** in the diagram). By the way of this association, if the type and the entity are known, one can determine the preferential proform or set of proforms used in the situation of the utterance.

Complete Model. These two models (signing space and schemes) are connected by the intermediary of the entities and the loci. An Entity is associated to a given Locus at a given moment of the situation, and *vice versa*.
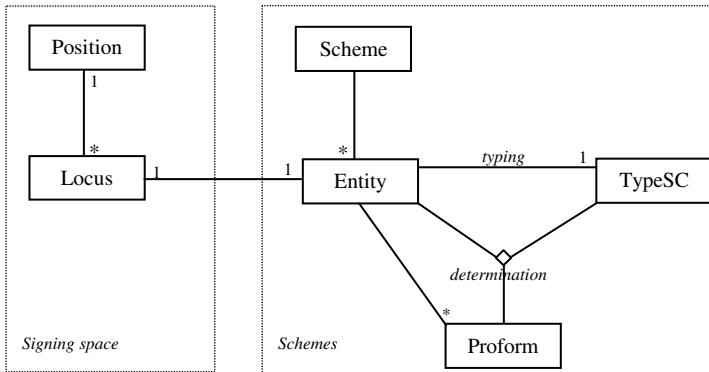
Fig. 2. Class diagram explaining the different concepts and their relations

We want this model to be independent of the computational application considered. This kind of model could be implemented in the context of recognition [28] or analysis systems [16] as a high control level, but this will not be described in this paper. The following section shows how it is used in an application which performs automatic generation of simple FSL utterances.

4.2 Automatic Generation of Utterance Description

To illustrate the principles of our modelling, we have developed a first prototype [29] with the aim to automatically produce simple isolated utterances such as "the cat is in the car" without having to specify the proforms or pointing gestures used and their spatialisation in the signing space.

The input is a triplet (entity1, relation, entity2), representing a spatialised relation *relation* between a localised entity *entity1* and a landmark entity *entity2*. The output is the animated utterance performed by a signing avatar (see Figure 3). The application is based on three processes that are very quickly described in the following.

1. In the first process, the scheme corresponding to the input is computed. This process uses a knowledge base dedicated to the relations. Each relation is associated to one or more operators and the way the relation is performed, by using proforms or pointing gestures. The following lines show an example for the relation 'in':

```
<relation nom="in">
 <localise val="prof"/>
 <repere val="prof"/>
<op-rep>
 <IN/>
</op-rep>
</relation>
```

2. In the second process, the sequence of gestural units corresponding to the scheme is computed. A set of instructions corresponding to the scheme operatos allows us to build the structure of the utterance (for the moment, only one structure is considered in the prototype), that is the list of gestural units composing the utterance.

For each gestural unit, a complete set of parameters must be computed (configuration, orientation, location, movement, gaze, etc). The needed places are chosen in the signing space in order to complete the description of the gaze direction, pointing gesture direction and proform location. The other gesture parameters are extracts from two knowledge bases:

- The first one is dedicated to the entities. For each entity, and for each possible semantico-cognitive type, a list of proforms and their possible realisations in space (orientations) is given.
- The second one lists the constraints in the realisation of the proforms, depending on the relation.

The whole description of the FSL utterance is stored in an XML file, with the different values for the different parameters.

3. Finally, this file is applied to our animation tool, which produces an OpenGl window showing the animated avatar, by reading prerecorded files describing each gestural units.

Fig. 3. Spatial relation between the cat and the car performed by the signing avatar

5   Conclusion and Perspectives

We describe in this paper a computational modelling of the formal model proposed in [18]. It is based on the modelling of signing space structure, with an emphasis on the concept of locus which are used to express the spatialised semantic relations, and also on the modelling of the relations, semantico-cognitive types, operators, entities and proforms as specified in the formal model. This model answers well our objective, which is to represent, independently of the application concerned, utterances expressing spatialised semantic relations between two entities.

In order to evaluate the model relevance, we have implemented it within an application of simple FSL utterance generation, performed by a signing avatar. These productions will be submitted in the next steps by native signers to evaluate the model capacities.

In the future, this model will have to be extended to make it possible to represent more complexe situation and successions of situations. It is also planned to integrate our model with the one proposed in [26] Thus, the proposition presented here is only a first step towards a more complete model.

References

1. Cuxac C.: La langue des signes française ; les voies de l'iconicité. In : Faits de Langues 15/16, Ophrys Paris (2000).
2. Sallandre M-A.: Les unités du discours en Langue des Signes Française. Tentative de catégorisation dans le cadre d'une grammaire de l'iconicité. PhD thesis, Paris 8 University, France (2003).
3. Fusellier I.: Sémiogénèse des langues des signes. Étude des langues des signes primaires (LSP) pratiquées par des sourds brésiliens. PhD thesis, Paris 8 University, France (2004).
4. B. Garcia 1998. Paul Jouison. Écrits sur la Langue des Signes Française (LSF). Harmattan (in french).
5. Risler a.: Noms et verbes en langue des signes: Ancrage perceptif des catégories lexicales. In: Santi, Serge et al (eds): Oralité et gestualité. Communication multimodale, interaction. L'Harmattan Paris (1998) 125-130
6. Emmorey K.: Language, cognition and the brain : Insignts from sign language, Lawrence Erlbaum Associates (2001).
7. Liddell S.: Grammar, gesture and meaning in American Sign Language. Cambridge Univ. Press, Cambridge (2003).
8. Talmy, L.: The representation of Spatial Structure in Spoken ansd Signed Language. In: Emmorey, K. (ed), Perspectives on classifier constructions in sign language, Lawrence Erlbaum Associates (2003) 311-332.
9. Suszczanska N. et al.: Translating Polish texts into Sign Language in the TGT system. In: 20th IASTED Int. Multi-Conference (2002) 282-287.
10. Zhao L., Kipper K., Schuler W., Vogler C., Badler N.: A Machine Translation System from English to American Sign Language. In: Association for Machine Translation in the Americas (2000).
11. Speers d'A.L.: Representation of American Sign Language for Machine Translation. PhD Dissertation, Department of Linguistics, Georgetown University (2001).
12. Safar E., Marshall I.: Sign language translation via DRT and HPSG. In : 3rd Int. Conf. On intelligent text processing and computational linguistics, LNCS 2276, Springer. (2002).
13. Elliott R. et al : An overview of the SiGML Notation and SiGMLSigning Software System. In: Workshop on the Representation and Processing of Signed Languages, LREC 2004, Portugal (2004).
14. Huenerfauth M.. Spatial Representation of Classifier Predicates for Machine Translation into American Sign Language. In: Workshop on the Representation and Processing of Signed Languages, LREC 2004, Portugal (2004).
15. Braffort A.: Reconnaissance et compréhension de gestes, application à la langue des signes. PhD thesis, Orsay University, France (1996).
16. Lenseigne B.: Intégration de connaissances linguistiques dans un système de vision, application à l'étude de la langue des signes. PhD thesis, Toulouse 3 University, France (2004).
17. Lejeune F., Braffort A.: Study on Semantic Representations of French Sign Language Sentences. In : Gesture and Sign Language based H-C Interaction, I. Wachsmuth & T. Sowa Eds, LNAI 2298 Springer (2002).
18. Lejeune F.: Analyse sémantico-cognitive d'énoncés en Langue des Signes Française pour une génération automatique de séquences gestuelles. PhD thesis, Orsay University, France (2004).
19. Desclés J.-P. : Les prépositions, relateurs ou opérateurs de repérage. In: *Colloque les relations,* Lille 3 University (1998).
20. Moody B.: La Langue des signes française, Tomes 1, 2 et 3,. IVT Paris 1998.

21. Crasborn O., Van der Hulst H., Van der Kooij E.: SignPhon: A phonological database fir sign languages. In: Sign Language and Linguistics, vol. 4 n°1, John Benjamins Pub. (2002) 215-228.
22. Prillwitz S. et al : HamNoSys version 2.0 ; HamburgNotation System for Sign Languages. An introduction guide. In: International Studies on Sign Languages and Communication of the deaf 5. Hamburg Signum (1989).
23. www.signwriting.org
24. Lebourque T. 1998. Spécification et génération de gestes naturels. Application à la Langue des Signes Française. PhD thesis Orsay University, France (in french).
25. Losson O. 2000. Modélisation du geste communicatif et réalisation d'un signeur virtuel de phrases en langue des signes française. PhD thesis, Lille 1 University, France (in french).
26. Lenseigne B. & Dalle P.: Using Signing Space as a Representation for Sign Language Processing. In: The 6th International Workshop on Gesture in Human-Computer Interaction and Simulation, France (2005).
27. Lenseigne B. & Dalle P.: Modélisation de l'espace discursif pour l'analyse de la langue des signes. In : Workshop on Traitement Automatique des Langues des Signes, TALN 2005, France (2005).
28. Bossard B. Braffort A. Jardino M.: Some issues in Sign Language processing. In: Gesture-based communication in human-computer interactopn, LNAI 2915, Springer (2004).
29. Braffort A. Bossard B. Segouat J. Bolot L. & Lejeune F.: Modélisation des relations spatiales en langue des signes française. In : Workshop on Traitement Automatique des Langues des Signes, TALN 2005, France (2005).

Automatic Generation of German Sign Language Glosses from German Words

Jan Bungeroth and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department,
RWTH Aachen University, D-52056 Aachen, Germany
`{bungeroth, ney}@informatik.rwth-aachen.de`

Abstract. In our paper we present a method for the automatic genera-
tion of single German Sign Language glosses from German words. Glosses
are often used as a textual description of signs when transcribing Sign
Language video data. For a machine translation system from German to
German Sign Language we apply glosses as an intermediate notational
system. Then the automatic generation from given German words is pre-
sented. This novel approach takes the orthographic similarities between
glosses and written words into account. The obtained experimental re-
sults show the feasibility of our methods for word classes like adverbs,
adjectives and verbs with up to 80% correctly generated glosses.

1 Introduction

In the field of automatic translation, significant progress has been made by using
statistical methods. This was successfully applied to many language pairs where
large amounts of data are available in the form of bilingual corpora. Using statis-
tical machine translation (SMT) for Sign Languages would require such corpora
too. Unfortunately only few data is available.

Addressing this data scarceness by using glosses as an intermediate Sign Lan-
guage notation, we provide a new approach to a Sign Language translation sys-
tem. The method presented in our paper is the first to examine the automatic
generation of glosses for German Sign Language (DGS) from German words. It
makes use of German base forms and a small bilingual corpus. We show how the
glosses are generated and give results for the different word classes. Our results
will show which word classes (e.g. adverbs) perform better than others.

2 Notation

For storing and processing Sign Language, a textual representation of the signs
is needed. While there are several notation systems covering different linguistic
aspects, we focus on the so called gloss notation. Glosses are widely used for
transcribing Sign Language video sequences.

In our work, a gloss is a word describing the content of a sign written with
capital letters. Additional markings are used for representing the facial expres-
sions. Unfortunately, no standard convention for glosses has been defined yet.

Table 1. Example glosses for DGS signs and their English translations

$\overline{\text{SITZEN}}^{\text{neg}}$	GELD+MÜNZEN	X-location
not sitting	money coins	reference to the location X

Furthermore, the manual annotation of Sign Language videos is a difficult task, so notation variations within one corpus are often a common problem.

In this work, the gloss notation basically follows the definitions as used in [1]. Additionally, compound nouns are separated with a plus if they are signed separately, and references to locations in signing space, signed with the hands, are given as an X with the location name.

Table 1 shows example glosses representing DGS signs. The glosses, retrieved from DGS video sequences, are given with their English translation.

3   Translation System

A complete Sign Language translation system, capable of generating Sign Language output from spoken input and for generated speech from recognized Sign Language, was proposed in [2].

The system propagates the use of a gloss notation for the corpus-based learning mechanisms. The input sentence (e.g. German) will be translated into glosses which are reordered according to the Sign Language grammar (e.g. DGS grammar). The corresponding animation performed by an avatar, that is a virtual signer, can be looked up in lexicons. Unknown glosses are still useful, as they can be finger-spelled.

4   Corpus

Bilingual Sign Language corpora are still rare, as the consistent annotation of videos is difficult. The available corpora are limited to a few hundred sentences, often taken from different domains. The European Cultural Heritage Online (ECHO) project [3] hosts a number of well annotated, small corpora from various Sign Languages like Swedish Sign Language (SSL), Dutch Sign Language (NGT) and British Sign Language (BSL). Furthermore, ECHO also published guidelines for annotation [4] and suitable software.

For our experiments we rely on a bilingual corpus, from the DESIRE team [5] for DGS and German consisting of 1399 sentences after pre-processing. Table 2 shows the corpus statistics where singletons are words occurring only once. Due

Table 2. DESIRE corpus statistics

	DGS	German
no. of sentence pairs	1399	
no. of running words	5480	8888
no. of distinct words	2531	2081
no. of singleton words	1887	1379

to the high number of singletons, this corpus is unsuitable for the immediate training in a SMT system.

Unfortunately, several sentences in the corpus use inconsistent annotation. Also some notations had to be changed. The altered notation will be used for testing the generated glosses later.

5 Gloss Generation

Obtaining a DGS gloss from a given German word is possible because the notation of the DGS sign is described with one or more German words. This similarity is also dependent on the semantic context of the DGS sentence and it's grammar. We can therefore expect words from some word classes to be generated better than those of others classes. Thus analyzing the word class of the German word is one basic idea of gloss generation.

For this analysis we rely on the commercially available analyzer by Lingsoft[1]. It writes the corresponding morpho-syntactical information of a German sentence to a file.

As an example, we look at the German sentence "Ich mag keine Nudeln." (*I don't like noodles.*). First we extract the base forms and process them for obtaining gloss-like words. That is, special symbols are removed and the obtained words are capitalized. Here, the resulting glosses would be: ICH, MÖGEN, KEIN, NUDEL. We then extract the word classes from this output. In this example that is pronoun (PRON), verb (V), determiner (DET) and noun (S). Note that an ambiguous word can have different interpretations.

Table 3 shows a further example sentence, where the German words are transformed to glosses.

Table 3. Example gloss generation

German	Ich kaufe heute ein neues Auto.
German base forms	ich kaufen heute ein neu Auto
Correct glosses	ICH KAUFEN HEUTE NEU AUTO
Correct DGS	HEUTE NEU AUTO KAUFEN
English	Today I buy a new car.

6 Results

For our experiments we extracted all the base forms of the German sentences in the DESIRE corpus. From these we generated the glosses using the methods described above. The resulting glosses were then compared with the DGS lexicon extracted from the DGS part of the corpus. All generated words were sorted according to their extracted base form. As mentioned in the last section about preprocessing, markings were handled as separate words.

With no further pre-processing we already achieved 55.7% correct matches overall. When looking at the distinct word classes different matching rates were

---

[1] http://www.lingsoft.fi

Table 4. Automatic gloss generation for different word categories

	NOUNS	VERBS	ADJ	ADV	PREP	PRON	CONJ	ART
no. of running words	940	924	304	321	248	598	67	275
no. of distinct words	710	210	135	61	35	30	11	10
correct glosses [in %]	52.1	67.1	72.6	81.0	48.6	46.7	0.0	0.0

found. Especially adverbs, adjectives and verbs could be generated easily and with a high compliance. Nouns were only generated correctly below average (52.1%). This is explained by a high number of compound nouns that are concatenated differently in DGS than in German. Also synonymous nouns are often used for the DGS transcription, so the generated gloss might be correct but not part of the lexicon. Further investigation on different corpora is necessary for noun generation.

On the other hand, the lack of conjunctions and articles is no surprise as words from these categories are rarely or even never used by signers in DGS. Preposition and pronouns should be handled with care, as those are often used in German, but in DGS they are often substituted by classifier predicates.

7   Summary and Outlook

We described how to generate single Sign Language glosses from given words. This method will be embedded into a complete translation system as described in this paper. The necessary corpus preparation was introduced as well as an overview of the gloss notation.

The generation process itself, where glosses are derived from the base form words, will assist the translation system. From the observed results we conclude to introduce automatic gloss generation for adjectives, adverbs and verbs. It should be possible to alter nouns during preprocessing for obtaining better results on this word class too. This will be addressed on other corpora as our next step towards automatic Sign Language translation.

References

1. C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, and R.G. Lee. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure.* MIT Press, Cambridge MA, 2000.
2. J. Bungeroth, and H. Ney. Statistical Sign Language Translation. *Workshop on Representation and Processing of Sign Languages, 4th International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 105–108, Lisbon, Portugal, May 2004.
3. O. Crasborn, E. van der Kooij, and J. Mesch. European Cultural Heritage Online (ECHO): Publishing sign language data on the internet. *8th conference on Theoretical Issues in Sign Language Research (TISLR8)*, Barcelona, October 2004.
4. A. Nonhebel, O. Crasborn, and E. van der Kooij. *Sign language transcription conventions for the ECHO project.* ECHO Project, 20 January 2004. Radboud University Nijmegen.
5. DESIRE. *DGS-Phrasensammlung.* Microbooks, Aachen, 1998.

French Sign Language Processing: Verb Agreement

Loïc Kervajan[1], Emilie Guimier De Neef[2], and Jean Véronis[1]

[1] DELIC team, Provence University, Aix-en-Provence 29,
avenue Robert Schuman, 13621 Aix-en-Provence Cedex 1, France
`loickervajan@wanadoo.fr, jean.veronis@up.univ-mrs.fr`
[2] France Télécom R&D/TECH/EASY/LN, Lannion,
avenue Pierre Marzin, 22307 Lannion Cedex
`emilie.guimierdeneef@francetelecom.com`

Abstract. In this paper, we propose an approach for the representation of the relationship between verbs and actors in French Sign Language. This proposal comes from the results of an experiment conducted in the France Télécom R&D's Natural Language laboratory using TiLT (Linguistic Treatment of Texts), an automatic syntactic analyser and generator. The aim of this approach is to develop a model that will be used to animate an avatar through a previous computational linguistics treatment, respecting French Sign Language as a proper human language.

Keywords: French Sign Language, verb typology, morphology, nominal classes, agreement, computational sciences.

1 Introduction

The present document describes the result of an experiment carried out as a collaborative work between France Télécom R&D's Natural Language laboratory and the DELIC (DEscription LInguistique sur Corpus: corpus-based linguistic description) team of Provence University. The main purpose of this experiment was to confirm that French Sign Language (FSL) could be computer-processed by adapting at a small cost a system originally developed for written and spoken languages, namely the TiLT syntactic analyser and generator, which is already operational for a number of languages such as French, German, English and Arabic. The problem tackled the representation of verbs and their arguments in the computational system, and the corresponding agreement.

2 Morphological Description of Verbs

Our machine representation is based on a typology that classifies verbs according to the way their morphology varies with inflexion, using affixes. We distinguish three types of affixes: **prefix**, **suffix** and **transfix**. Those terms must be understood through the usual way flexional languages build their flexions using

affixes in order to describe syntactic relations. But they also must be understood through the specific way sign languages use space in order to realize the signed sentence. For instance the prefix of the verb refers to the place where the movement starts, the suffix refers to where the movement ends. Prefixes and suffixes as places are named "loci", a term used by A. Vercaigne-Ménard and D. Pinsonneault [1]. A locus is a subset of the space referring to an element of the discourse formerly associated with this place or, if not formerly specified, referring to conventional pronouns. The transfix is a term corresponding to another specificity of sign language which can superpose elements. For instance, the transfix is realized at all times during the articulation of the verb, prefix and suffix included. In other words, a transfix affects the verb in such a way that the verb "takes its appearance". In verbal morphology, the transfix is always a gesture profom. "Gesture proforms" (see Slobin [2]) have long been called "classifiers". But on a purely syntactic point of view, proforms are manipulated like pronouns, referring to objects (or persons) already enunciated in the discourse. Furthermore, proform is specific in linguistics since it can be adapted in function of the objects to which it refers (see Cuxac [3]). Figure 1 shows an example of the relation between verb an noun, in the which:

  - [**give**] and [**glass**] are the word roots
  - "**I Locus**" is the place where the verb begins (the body of the hearer)
  - "**you Locus**" is the place where the verb finishes (where the co-speaker stands)
  - "**C Proforme**" is the gesture proform referring to the glass, by which the verb is realized.

In our dictionary each noun is associated to a list of gesture proforms. The list comes from the systematic observation of the realization of verbs by signing
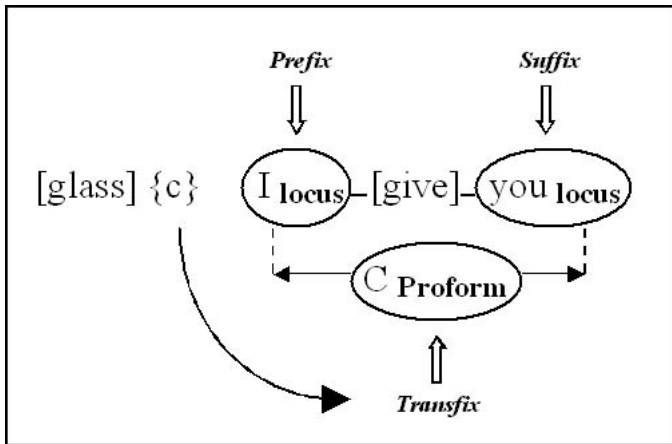
Fig. 1. "I give you a glass"

deaf people. The agreement matches if a gesture proform is licensed by the verb through a transfix. In our example, one of the gesture proforms of [**glass**] is "**c**" and [**give**] licenses "**c**" to be constructed. As several proforms may be associated to nouns and verbs, the system will have to use information given by contextual entries required to build the semantic graph.

This was an example of verbs whose morphology varies with the three affixes. In that case, the first locus refers to the subject, the second to the oblique and the transfix to the object. Thus this verb admits an agreement with three arguments: subject, oblique and object. Let us see some other types of agreement:

– [**dream**] does not vary even if it has arguments (at least a subject, and an object)

   [**he**] [**dreamed**] *(he was in Paris)*[1]   He dreamed *(he was in Paris)*

As the verb does not specify the subject through its morphology, [**he**] has to be articulated as an independent gesture.

– [**tell**] only varies in subject and oblique:

   I-[tell]-you *(he is gone)*   I tell you *(that he is gone)*

Some verbs have a morphology that varies with prefixes and suffixes, but whose loci refer to locations:

– [**take the plane**] varies in locations, the other arguments being independent:

   [**Paris**](**on locA**) [**L.A.**](**on locB**) [**you**] **locA-[take the plane]-locB**
                                        you take the plane from Paris to L.A.

3   Codification for Computational Treatment

Each verb is associated to a morphological code "V_" followed by three letters representing the morphological variations with affixes in the following order: prefix, suffix and transfix.

Each letter encodes the morphological variation with:

locus	L
gesture proform	P
nothing (invariant)	X

For instance,

– "to dream" is encoded V_XXX since it doesn't vary
– "to tell" is encoded V_LLX since its prefix and suffix both vary
– "to give" is encoded V_LLP since its prefix, suffix and transfix all vary

---

[1] The part in parentheses and italic is not treated since it does not belong to the demonstration.

The inflectional model for verbs combines the different forms of its various affixes:

Table 1. Variations for [give]

lemma	prefix	postfix	transfixe
V-LLC	Locus	Locus	Proform
give	I	I	c
	you	you	o
	he	he	open_duckbill
	Pointer	Pointer	closed_duckbill
			clamp
			open_clamp
			. . .

4   Conclusion

On the basis of the representations we have defined, which rely on a typology of verbs and nominal classes, the TiLT syntactic parser is able to build grammatical dependency trees for FSL. So far, our study seems to indicate that a slight adaptation of existing tools, provided that they are based on adequate sign language description, is sufficient to carry out some aspects of computer processing.

Furthermore, this representation of the verbs and the way they are constructed with affixes is a good way to think of the link between the syntactic module and an avatar. Indeed, this codification gives in itself some indications on interrelating places and movements within the grammatical space of elocution.

References

1. Vercaingne-Ménard, A., & D. Pinsonneault (1996). "L'établissement de la référence en LSQ : les loci spatiaux et digitaux". In C. Dubuisson et D. Bouchard (dir.), Spécificités de la recherche linguistique sur les langues signées, Montréal : Les cahiers scientifiques de l'Acfas, no 89, pp 61–74.
2. Slobin, D. I & al. (2001). A cognitive/functional perspective on the acquisition of "classifiers". In Emmorey, K. (ed.), Perspective on Classifier Constructions in Sign Languages. Paper presented to conference "classifier constructions in Sign Languages", La Jolla, CA. (2000, April)
3. Cuxac C. (2000). La langue des signes Française, les Voies de l'Iconicité, Faits de langue n15–16, Paris: Ophrys.

Re-sampling for Chinese Sign Language Recognition

Chunli Wang[1,2], Xilin Chen[1], and Wen Gao[1]

[1] Institute of Computing Technology, Chinese Academy of Science, Beijing 100080, China
[2] Department of Computer Science and Engineering,
School of Electronic and Information Engineering,
Dalian University of Technology, Dalian 116023, China
{clwang, xlchen, wgao}@jdl.ac.cn

Abstract. In Sign Language recognition, one of the problems is to collect enough data. Data collection for both training and testing is a laborious but necessary step. Almost all of the statistical methods used in Sign Language Recognition suffer from this problem. Inspired by the crossover and mutation of genetic algorithms, this paper presents a method to enlarge Chinese Sign language database through re-sampling from existing sign samples. Two initial samples of the same sign are regarded as parents. They can reproduce their children by crossover. To verify the effectiveness of the proposed method, some experiments are carried out on a vocabulary with 350 static signs. Each sign has 4 samples. Three samples are used to be the original generation. These three original samples and their offspring are used to construct the training set, and the remaining sample is used for testing. The experimental results show that the data generated by the proposed method are effective.

1 Introduction

Hand gesture recognition, which contributes to a natural man-machine interface, is still a challenging problem. Closely related to the realm of gesture recognition is that of sign language recognition. Sign language is one of the most natural means of exchanging information for the hearing impaired people. It is a kind of visual language via hand and arm movements accompanying facial expression and lip motion. The aim of sign language recognition is to provide an efficient and accurate mechanism to translate sign language into text or speech.

The reports about gesture recognition began to appear at the end of 80's. T.Starner [1] achieved a correct rate of 91.3% for 40 signs based on the image. By imposing a strict grammar on this system, the accuracy rates in excess of 99% were possible with real-time performance. Fels and Hinton [2][3] developed a system using a VPL DataGlove Mark II with a Polhemus tracker as input devices. Neural network was employed for classifying hand gestures. Y. Nam and K.Y. Wohn [4] used three–dimensional data as input to Hidden Markov Models (HMMs) for continuous recognition of a very small set of gestures. R.H.Liang and M. Ouhyoung [5] used HMM for continuous recognition of Tainwan Sign language with a vocabulary between 71 and 250 signs by using Dataglove as input devices. HMMs were also

adopted by Kisti Grobel and Marcell Assan to recognize isolated signs collected from video recordings of signers wearing colored gloves, and 91.3% accuracy out of a 262-sign vocabulary was reported [6]. C.Vogler and D.Metaxas [7] described an approach to continuous, whole-sentence ASL recognition, in which phonemes instead of whole signs were used as the basic units. They experimented with 22 words and achieved similar recognition rates with phoneme-based and word-based approaches. Wen Gao[8] proposed a Chinese Sign language recognition system with a vocabulary of 1064 signs. The recognition accuracy is about 93.2%. C. Wang [9] realized a Chinese Sign Language (CSL) recognition system with a vocabulary of 5100 signs.

For signer-independent recognition, Vamplew [10] reported the SLARTI sign language recognition system with an accuracy of around 94% on the signers used in training, and about 85% for other signers. It used a modular architecture consisting of multiple feature-recognition neural networks and a nearest-neighbor classifier to recognize 52 Australian sign language hand gestures. All of the feature-extraction networks were trained on examples gathered from 4 signers, and tested on both fresh examples from the same signers and examples from 3 other signers. Akyol and Canzler [11] proposed an information terminal that can recognize 16 signs of German Sign Language from video sequences. 7 persons were taken for training the HMMs and the other three for testing. The recognition rate is 94%.

Up to now, one of the problems in Sign Language recognition is to collect enough data. Data collection for both training and testing is a laborious but necessary step. Almost all of the statistical methods used in Sign Language Recognition suffer from this problem. However, sign language data cannot be gotten as easily as speech data. We must invite the special persons to perform the signs. The lack of the data makes the research, especially the large vocabulary signer-independent recognition, very difficult. In face detection and recognition field, researchers employ some methods to generate new samples to swell the face database [12]. This paper focuses on this problem. Re-sampling is presented to enlarge the sign language database. Inspired by genetic algorithms, the ideas of crossover and mutation are used to generate more samples from existing ones. Each sign is composed of limited types of components, such as hand shape, position and orientation, which are independent of each other. Two initial samples of the same sign cross at one and only one component to generate two children.

The rest of this paper is organized as follows. In Sect. 2, the re-sampling method based on genetic algorithms is proposed. In Sect. 3, the experimental results are reported. Finally in Sect. 4, we give the conclusions.

2   Sign Re-sampling

In order to get more training data, we can generate new samples from the existing ones. The ideas of genetic algorithms, namely crossover and mutation, can be employed.

2.1   Representing a Sign

Two CyberGlove and a Pohelmus 3-D tracker with three receivers positioned on the wrist of CyberGlove and the back are used as input device in this system. The input equipments are shown in Fig. 1.

Fig. 1. The Dataglove and the 3-D tracker used in our system. Three receivers are fixed on two hands and the back respectively.

Each sample of a sign is a sequence of frames. The number of frames is from 10 to 20. One frame of raw gesture data, which in our system are obtained from 36 sensors on two datagloves, and three receivers mounted on the datagloves and the body, are formed as 48-dimensional vector. An algorithm based on geometrical analysis for the purpose of extracting invariant feature to signer position is employed [9]. Each element value is normalized to ensure its range 0-1.

Each hand is represented by 24 dimensions data. 18 dimensions data represent the hand shape, 3 dimensions data represent the position of hand, and 3 dimensions data represent the orientation of hand. We can split up one sign into a number of channels. Each channel can be considered as a gene of the sign. Inspired by the genetic algorithms, crossover and mutation can be used to generate new samples of the sign. Intuitively, we may adopt the following four splitting strategies.

1. S = {Left Hand, Right Hand}, Channel Number = 2.
2. S = {Position&Orientation, Left Hand Shape, Right Hand Shape}, Channel Number = 3.
3. S = {Left Position&Orientation, Left Hand Shape, Right Position&Orientation, Right Hand Shape}, Channel Number = 4.
4. S = {Left Position, Left Orientation, Left Hand Shape, Right Position, Right Orientation, Right Hand Shape}, Channel Number = 6.

In the following we address how to generate new samples in detail according to the Strategy-2. The procedures according to other strategies are similar. In Sect. 4, the performances of these four splitting strategies are evaluated.

2.2 Generating New Samples

Genetic algorithms take their analogy from nature. Two initial samples of the same sign are regarded as parents. They can reproduce their children by crossover and mutation.

Each sign is composed of limited types of components, such as hand shape, position and orientation. We can split up one sample into three channels, namely Position & Orientation (P&O), Left Hand Shape (LH), and Right Hand Shape (RH). These three parts of the same sign in different samples may be variable. One sample gives one demonstration of each part. There are two samples of "Australia" shown in Fig. 2.

Fig. 2. Two samples of the sign "Australia". The hand shapes, positions and orientations of these two samples are different.

S_1 and S_2 denote these two samples of the same sign "Australia". $S_1 = \{P\&O_1, LH_1, RH_1\}$, $S_2 = \{P\&O_2, LH_2, RH_2\}$. The hand shape, orientation and position of the first sample are different from that of the second one. All of them are correct for this sign. So a gesture with the position & orientation of the first sample and the hand shape of the second one, namely $S = \{P\&O_1, LH_2, RH_2\}$, is a possible sample of the sign but is different from S_1 and S_2.

So the new sample S may not match the model well and may not be recognized correctly. If we re-combine these parts from different samples, the model trained by them can have better generalization performance.

Two samples of the same sign are picked up from the initial training set randomly. Each sample is a sequence of frames. In order to cross, the length of Parent2 should be warped to that of Parent1. Each frame is divided into three parts: position & orientation, left hand shape and right hand shape. Every pair of parent crosses at one and only one part, that is, all frames of the parent cross over at the same part. The parent can cross three times and each crossover operator can generate two children, so two samples can generate 6 new samples. The process of crossover is shown in Fig. 3.

During the simulation process, the signs are also mutated. When two parents cross, the part from Parent2 is mutated randomly within the limited scale. The variety range can be learned from the training set. The variance in each dimension is calculated from all the frames of all the samples and is used to control the extent of mutation. Besides, the description of signs in dictionary is referenced, too. For example, if the hand shapes of one sign are very important, the mutation can cover the variations in position and orientation. Because we do not design an evaluation function to judge the fitness of a new sample, the mutation is limited to a small range. So the mutation operator has less effect on the results than crossover operator.
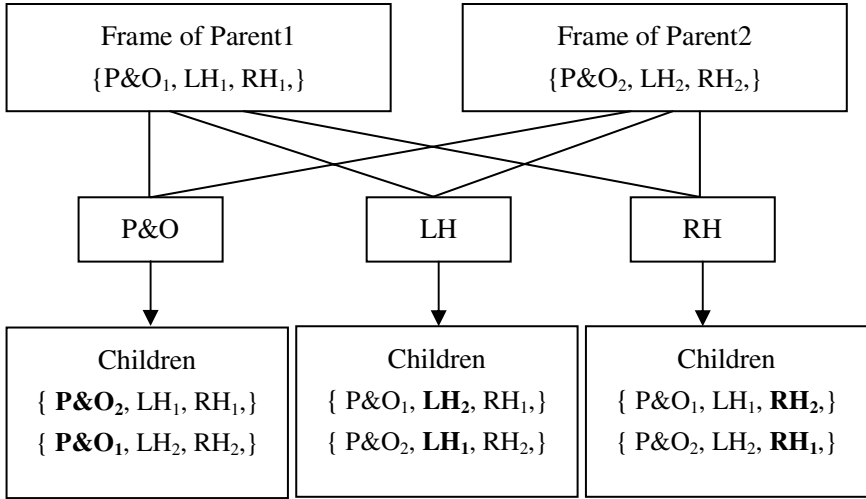
Fig. 3. Crossover operator during the re-sampling. Each frame is broken down into three independent parts. During a crossover, one and only one part is exchanged. All frames of the parent cross over at the same part.

To verify the effects of the proposed method, we compare it with Parallel HMMs (PaHMMs).

2.3 PaHMMs

As mentioned in Sect 2.1, if the data of the channel 1 of a test sample are similar to a training sample of the same sign, while the data of channel 2 are similar to another training one, the test sample may be different from all the training samples and is probably recognized as another sign. To verify the effects of the above method, we assume the other methods that can resolve this situation. Christian Vogler [13] proposed a framework of PaHMMs, which builds a model for each channel respectively and can probably resolve this problem.

Corresponding to the four strategies given in Sect. 2.1, we design 4 kinds of PaHMMs as follows:

1. PaHMM-CN2 models 2 channels with 2 independent HMMs. The two channels are: Left Hand and Right Hand.
2. PaHMM-CN3 models 3 channels with 3 independent HMMs. The three channels are: Position&Orientation, Left Hand Shape, and Right Hand Shape.
3. PaHMM-CN4 models 4 channels with 4 independent HMMs. The four channels are: Left Position&Orientation, Left Hand Shape, Right Position&Orientation, and Right Hand Shape.
4. PaHMM-CN6 models 6 channels with 6 independent HMMs. The six channels are: Left Position, Left Orientation, Left Hand Shape, Right Position, Right Orientation, and Right Hand Shape.

The PaHMM-CN3 is given in Fig. 4. The others are similar.
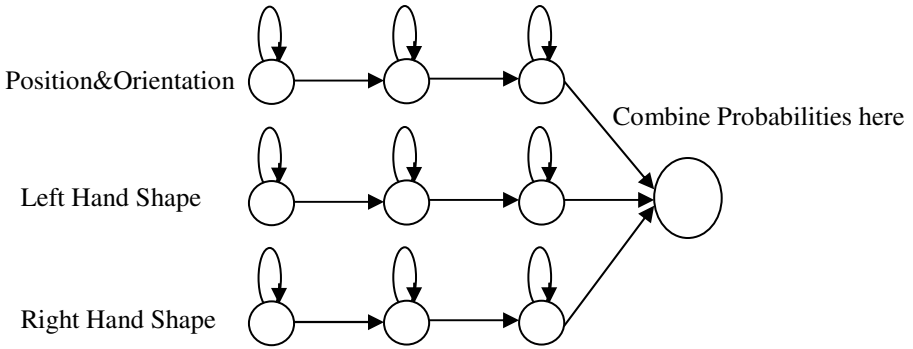
Fig. 4. PaHMMs with 3 independent channels

In Sect. 4, the performances of these four kinds of PaHMMs are evaluated and the comparisons between the results of PaHMMs and HMMs trained by generated samples are reported.
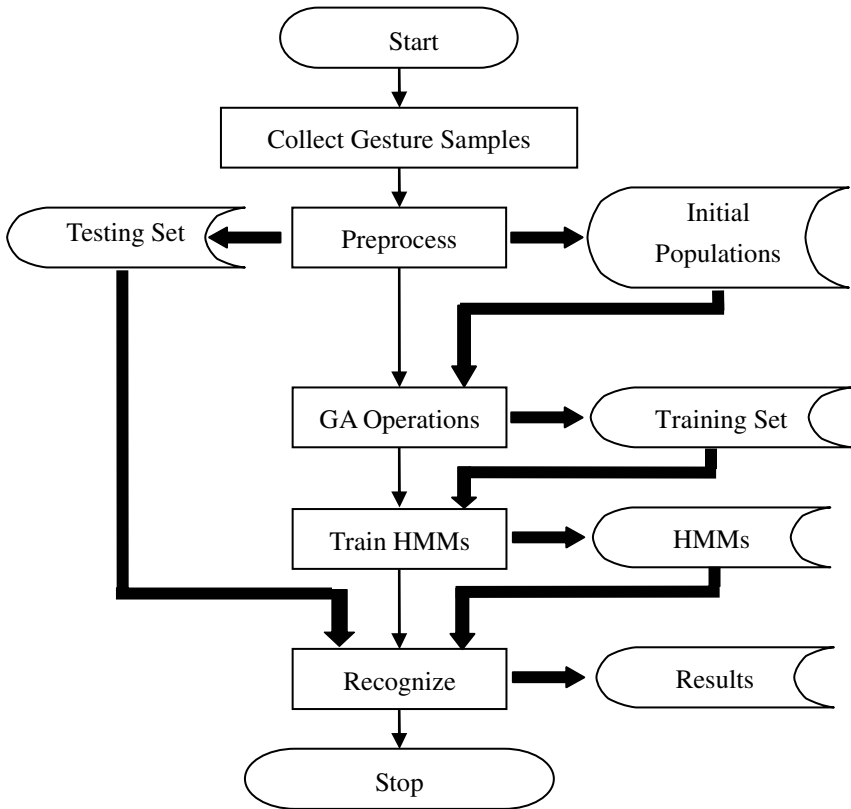
Fig. 5. Overview of the recognition system combining the proposed sign re-sampling method

2.4   Overview of the Recognition System

The framework of the recognition system combining the proposed sign re-sampling method is shown in Fig. 5. Firstly, the input data are preprocessed. The sign data collected by the gesture-input devices is fed into the feature extraction module, and then normalized. The dataset contains a small number of samples, that is, 4 samples for each sign. One sample of each sign is employed as the testing one. The others are used as the initial population to perform the GA operations. The initial samples and their children are utilized to train the HMMs. Because the number of training samples increases, the training time will increase. But the training time is seldom considered. The recognition time and accuracy of a system are more important. Viterbi is used in our system. The recognition process is the same as the regular one. The computational complexity of the decoding algorithm and the recognition time do not change.

3   Experiments

To verify the generalization capability of the proposed method, some experiments are performed. In order to make it simple to warp two sequences, we carry out experiments based on a vocabulary with 350 static signs, in which the hand shape, position and orientation change slightly. These signs are captured by the same signer. We invite a deaf signer to collect data for us. Each sign has 4 samples. The traditional leave-one-out cross-validation is employed. Three samples are used to construct the original training data, and the remaining one is used for testing. So there are four groups of training sets and test samples. The numbers of generated samples according to different strategies are shown in Table 1.

Table 1. The Numbers of Samples According to Different Strategies

	Original	Strategy-1	Strategy-2	Strategy-3	Strategy-4
New Samples	0	6	18	30	42
Training Samples	3	9	21	33	45

HMMs are trained based on different training sets. In our system, HMM is left-to-right model allowing possible skips. The number of states is set to be 3, and the number of mixture components is set to be 2. We fix the values of variances of covariance matrix. The variances of the feature data in the dimensions representing Position and Orientation are set to be 0.2 and those in the dimensions representing Hand Shape are set to be 0.1. These above values are obtained by experiments.

The recognition results according to different strategies are shown in Fig. 6. From Fig. 6, it can be seen that the results according to any strategy are better than those based on the original training data. Generating new samples improves the accuracy.
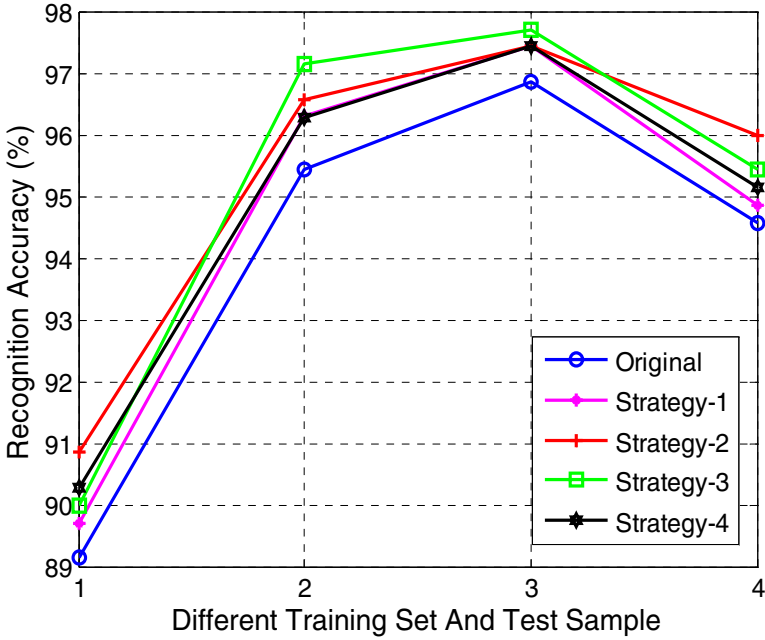
Fig. 6. Comparison of the recognition results based on the original training data and the generated training data according to different strategies. Leave-one-out cross-validation is employed.

The accuracy of the first group is lower than those of the others. The possible reasons are as follows. The test data in this group were collected for the first time. The signer was not very accustomed to perform gestures with unwieldy dataglove and tracker on body. Some signs are not up to the standard. Besides, when we collected data for the first time, some details of the input devices, such as the effective range of the tracker, are neglected. The data are somewhat affected.

Table 2 gives the average recognition rates of above 5 methods.

Table 2. The average recognition rates based on the original training set and four splitting strategies

Training Set	Original	Strategy-1	Strategy-2	Strategy-3	Strategy-4
Average Accuracy	**94.00%**	94.57%	**95.21%**	95.07%	94.78%

The average accuracy based on the original training data is 94%. Strategy-2 achieves the best accuracy of 95.21%. According to the relative accuracy improvement computing formula (1):

$$\Delta\lambda = \frac{\lambda_1 - \lambda_2}{1 - \lambda_2} \tag{1}$$

Table 3. The average recognition rates of HMMs trained by generated samples and PaHMMs with different channel number

Model	CN =2	CN =3	CN =4	CN =6
HMMs	94.57%	95.21%	95.07%	94.78%
PaHMMs	94.10%	94.36%	94.71%	94.57%
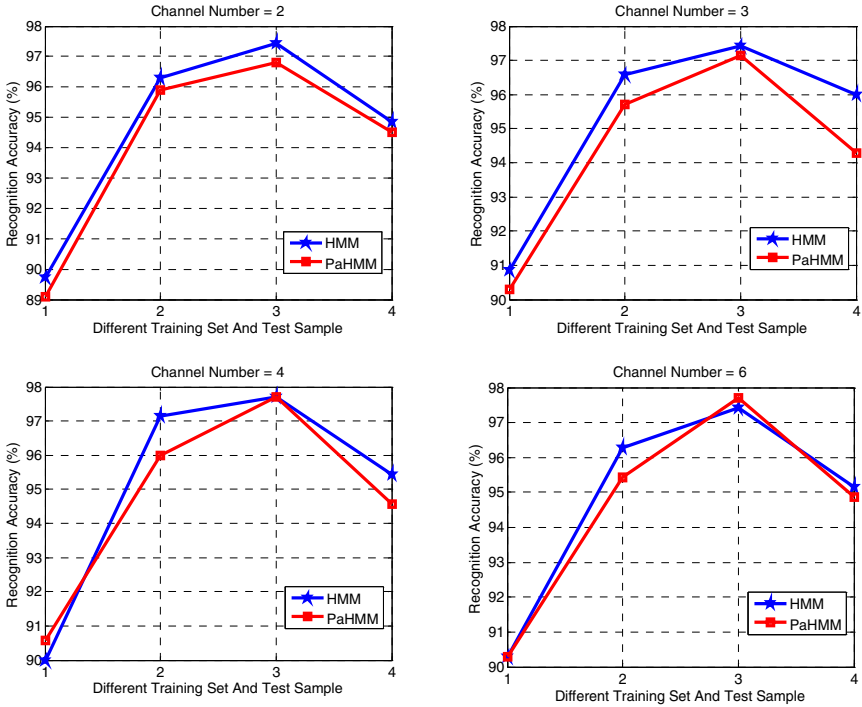
Fig. 7. The comparisons between HMMs trained by generated training sets and PaHMMs with different channel number based on cross-validation tests

The relative accuracy improvement of 20% is achieved. This result is very encouraging. The experimental results show that the data generated by the proposed method are effective.

The possible reasons for the above results are as follows. The generated new samples may be different from the original training samples but similar to the unknown test sample. So by this method, the system can get better generalization performance with the limited training data. According to Strategy-1, only few new samples are generated, which are not enough yet. So the improvement is limited. According to Strategy-4, the position and orientation are considered as different channels. But they are not absolutely independent from each other, so the crossover operator may generate unreasonable samples.

To verify the effects of the re-sampling method, we carry out some experiments on PaHMMs. Table 3 summarizes the average accuracies of HMMs trained by generated samples and PaHMMs with different channel number.

It can be seen that generating new samples may achieve better accuracy than PaHMMs. The recognition rates on cross-validation tests based on different channel number are given in Fig. 7.

4   Conclusions and Future Work

The re-sampling method based on the crossover and mutation of genetic algorithms is proposed to swell the sign language database and improve the recognition accuracy of gestures. The re-sampling is designed to generate a number of new samples, which are used to train HMMs, from the existing ones. Crossover is employed to simulate the procedure. Experiments conducted on a sign language database containing 350 static gestures (1440 gesture samples) show that the recognition accuracy is improved by applying the proposed method.

This idea can be used for dynamic gestures, too. But the procedure of crossover and mutation will be much more complicated. The parents should be warped by dynamic programming, such as DTW. Furthermore, there are more factors that can be mutated, for example, the scope or speed of action, the track of movement, etc. How to generate the gestures of other signers is the key to resolve the absence of data of the signer-independent sign language recognition systems.

Acknowledgment

References

1. Starner T., Weaver J., Pentland A.: Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. IEEE PAMI, Vol 20, Issue 12, Dec 1998, pages 1371-1375.
2. S.S.Fels, G.Hinton.: GloveTalk:A neural network interface between a DataDlove and a speech synthesizer. IEEE Transactions on Neural Networks, 4(1993):2-8.
3. S.Sidney Fels.: Glove –TalkII: Mapping hand gestures to speech using neural networks-An approach to building adaptive interfaces. PhD thesis, Computer Science Department, University of Torono, 1994.
4. Yanghee Nam, K. Y. Wohn.: Recognition of space-time hand-gestures using hidden Markov model. ACM Symposium on Virtual Reality Software and Technology, HongKong, July, 1996, pp51-58.
5. R.-H.Liang, M.Ouhyoung: A real-time continuous gesture recognition system for sign language. In Proceeding of the Third International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pages 558-565.
6. Kirsti Grobel, Marcell Assan: Isolated sign language recognition using hidden Markov models. In Proceedings of the International Conference of System,Man and Cybernetics,1996,pages 162-167.

7. ChristianVogler, Dimitris Metaxas: Toward scalability in ASL Recognition: Breaking Down Signs into Phonemes. In Proceedings of Gesture Workshop, Gif-sur-Yvette, France, 1999, pages 400-404.
8. Wen Gao, Jiyong Ma, Jiangqin Wu and Chunli Wang.: Large Vocabulary Sign Language Recognition Based on HMM/ANN/DP. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14, No. 5 (2000) 587-602.
9. Chunli Wang, Wen Gao, Jiyong Ma.: A Real-time Large Vocabulary Recognition System for Chinese Sign Language. Gesture and Sign Language in Human-Computer Interaction. London, UK, April 2001, 86-95.
10. Vamplew, P., Adams, A.: Recognition of Sign Language Gestures Using Neural Networks. Australian Journal of Intelligent Information Processing Systems, Vol. 5, No. 2, Winter 1998, 94-102.
11. Suat Akyol, Ulrich Canzler.: An information terminal using vision based sign language recognition. ITEA Workshop on Virtual Home Environments, 2002, 61-68.
12. Jie Chen, Xilin Chen, Wen Gao.: Expand Training Set for Face Detection by GA Re-sampling. the 6th IEEE International Conference on Automatic Face and Gesture Recognition(FG2004), Seoul, Korea, May17-19,2004, pp73-79
13. C. Vogler, D. Metaxas.: Handshapes and movements: Multiple-channel ASL recognition. Springer Lecture Notes in Artificial Intelligence 2915, 2004. Proceedings of the Gesture Workshop'03, Genova, Italy. pp. 247-258.

Pronunciation Clustering and Modeling of Variability for Appearance-Based Sign Language Recognition

Morteza Zahedi, Daniel Keysers, and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department,
RWTH Aachen University, D-52056 Aachen, Germany
{zahedi, keysers, ney}@informatik.rwth-aachen.de

Abstract. In this paper, we present a system for automatic sign language recognition of segmented words in American Sign Language (ASL). The system uses appearance-based features extracted directly from the frames captured by standard cameras without any special data acquisition tools. This means that we do not rely on complex preprocessing of the video signal or on an intermediate segmentation step that may produce errors. We introduce a database for ASL word recognition extracted from a publicly available set of video streams. One important property of this database is the large variability of the utterances for each word. To cope with this variability, we propose to model distinct pronunciations of each word using different clustering approaches. Automatic clustering of pronunciations improves the error rate of the system from 28.4% to 23.2%. To model global image transformations, the tangent distance is used within the Gaussian emission densities of the hidden Markov model classifier instead of the Euclidean distance. This approach can further reduce the error rate to 21.5%.

1 Introduction

In the domain of sign language recognition from video, most approaches try to segment and track the hands and head of the signer in a first step and subsequently extract a feature vector from these regions [1, 2, 3, 4]. Segmentation can be difficult because of possible occlusions between the hands and the head of the signer, noise or brisk movements. Many approaches therefore use special data acquisition tools like data gloves, colored gloves or wearable cameras. These special tools may be difficult to use in practical situations.

In this work, we introduce a database of video streams for American sign language (ASL) word recognition. The utterances are extracted from a publicly available database and can therefore be used by other research groups. This database, which we call 'BOSTON50', consists of 483 utterances of 50 words. One important property of this database is the large visual variability of utterances for each word. This database is therefore more difficult to recognize automatically than databases in which all utterances are signed uniformly. So far, this problem has not been dealt with sufficiently in the literature on sign language recognition.

To overcome these shortcomings we suggest the following novel approaches:

1. The system presented in this paper is designed to recognize sign language words using simple appearance-based features extracted directly from the frames which are captured by standard cameras without any special data acquisition tools. This means that we do not rely on complex preprocessing of the video signal or on an intermediate segmentation step that may produce errors.

2. Because of the high variability of utterances of the same class, we explicitly model different pronunciations of each word of the database. We employ and compare different clustering methods to determine the partitioning into pronunciations: manual clustering, k-means clustering, and hierarchical LBG-clustering. Manual clustering uses a hand-labeled partitioning of the utterances. The k-means algorithm is initialized with the number of clusters and manually selected seed utterances. The hierarchical LBG-clustering partitions the data automatically and only needs one parameter to control the coarseness of the clustering. The results obtained lead us to also consider a nearest neighbor classifier that performs surprisingly well.

3. To deal with the image variability, we model global affine transformations of the images using the tangent distance [6] within the Gaussian emission densities instead of the Euclidean distance.

In Sections 2 and 3, we introduce the database BOSTON50 and the appearance-based features used in the system, respectively. Section 4 describes the decision making and the hidden Markov model (HMM) classifier. Tangent distance and the way it is employed in the HMM is explained in Section 5. In Section 6, the different clustering methods and their properties are described. Finally, the experimental results and conclusions are discussed in Sections 7 and 8.

2   Database

The National Center for Sign Language and Gesture Resources of the Boston University has published a database of ASL sentences[1] [7]. It consists of 201 annotated video streams of ASL sentences. Although this database was not recorded primarily for image processing and recognition research, we considered it as a starting point for a recognition corpus because the data are available to other research groups and, thus, can be a basis for comparisons of different approaches.

The signing is captured simultaneously by four standard stationary cameras where three of them are black/white and the remaining one is a color camera. Two black/white cameras, directed towards the signer's face, form a stereo pair. Another camera is installed on the side of the signer. The color camera is placed between the cameras of the stereo pair and is zoomed to capture only the face of the signer. The movies are recorded at 30 frames per second and the size of the frames is 312×242 pixels. We use the published video streams at the same frame rate but extract the upper center part of size 195×165 pixels. (Parts of the bottom of the frames show some information about the frame and the left and right border of the frames are unused.)

---

[1] http://www.bu.edu/asllrp/ncslgr.html

Fig. 1. The signers as viewed from the two camera perspectives

To create our database for ASL word recognition which we call BOSTON50, we extracted 483 utterances of 50 words from this database as listed in the appendix along with the number of utterances of each word. The utterances of the sign language words are segmented within our group manually.

In the BOSTON50 database, there are three signers, one of them male and two female. The signers are dressed differently and the brightness of their clothes is different. We use the frames captured by two of the four cameras, one camera of the stereo camera pair in front of the signer and the lateral camera. Using both of the stereo cameras and the color camera may be useful in stereo and facial expression recognition, respectively. Both of the cameras used are in fixed positions and capture the videos simultaneously. The signers and the views of the cameras are shown in Figure 1.

3   Feature Extraction

In this section, we briefly introduce the appearance-based features used in our ASL word recognition system. In [5], we introduce different appearance-based features in more detail, including the original image, skin color intensity, and different kinds of first- and second-order derivatives. The results show that down-scaled original images extracted after skin intensity thresholding perform very well. According to these results we employ these features in the work presented here.

The definition of the features is based on basic methods of image processing. The features are directly extracted from the images of the video frames. We denote by $Y_t(i, j)$ the pixel intensity at position (i, j) in the frame t of a sequence, $t = 1, \ldots, T$.

To disregard background pixels, we use a simple intensity thresholding. This thresholding aims at extracting the hand and the head, which form brighter regions in the images. This approach is not a perfect segmentation and we cannot rely on it easily for tracking the hands because the output of the thresholding consists of the two hands, face and possibly some parts of the signer's clothes.

Fig. 2. Example of the features used by the classifier: original image (left), thresholded image (center), and down-scaled image (right)

$$X_t(i,j) = \begin{cases} Y_t(i,j) & : & Y_t(i,j) > \Theta \\ 0 & : & \text{otherwise} \end{cases} \tag{1}$$

Where $X_t(i,j)$ is an image frame at time t with the brightness threshold Θ.

We can transfer the matrix of an image to a vector x_t and use it as a feature vector. To decrease the size of the feature vector, we use the original image down-scaled to 13×11 pixels denoted by X'_t.

$$x_{t,d} = X'_t(i,j), \quad d = 13 \cdot j + i, \tag{2}$$

where $x_t = [x_{t,1}, ..., x_{t,d}, ..., x_{t,D}]$ is the feature vector at time t with the dimension $D = 143$.

Some examples of features after processing are shown in Figure 2. To increase the information extracted from the videos, we may use the frames of two cameras. One of the cameras is installed in front of the signer and the second one is fixed at one side. We concatenate the information of the frames captured simultaneously by these cameras. We weight the features extracted by the two cameras because there is more occlusion of the hands in the images captured by the lateral camera. According to experiments reported in [5], we weight the features of the front camera and lateral camera with the weights 0.38 and 0.62, respectively.

4   Decision Process

The decision making of our system employs HMMs to recognize the sign language words[2]. This approach is inspired by the success of the application of HMMs in speech recognition [8] and also most sign language recognition systems [1, 2, 3, 4, 5]. The recognition of sign language words is similar to spoken word recognition in the modelling of sequential samples. The topology of the HMM used is shown in Figure 3. There is a transition loop at each state and the maximum allowed transition is set to two, which means that, at most, one state can be skipped. We consider one HMM for each word $w = 1, ..., W$. The basic decision rule used for the classification of $x_1^T = x_1, ..., x_t, ... x_T$ is:

---

[2] Some of the code used in feature extraction and decision making is based on the LTI library that is available under the terms of the GNU Lesser General Public License at http://ltilib.sourceforge.net.
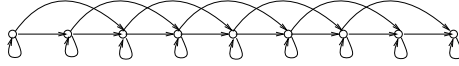
Fig. 3. The topology of the employed HMM

$$x_1^T \longrightarrow r(x_1^T) = \arg\max_w \left(Pr(w|x_1^T) \right)$$
$$= \arg\max_w \left(Pr(w) \cdot Pr(x_1^T|w) \right), \tag{3}$$

where $Pr(w)$ is the prior probability of class w, and $Pr(x_1^T|w)$ is the class conditional probability of x_1^T given class w. The $Pr(x_1^T|w)$ is defined as:

$$Pr(x_1^T|w) = \max_{s_1^T} \prod_{t=1}^{T} Pr(s_t|s_{t-1}, w) \cdot Pr(x_t|s_t, w), \tag{4}$$

where s_1^T is the sequence of states, and $Pr(s_t|s_{t-1}, w)$ and $Pr(x_t|s_t, w)$ are the transition probability and emission probability, respectively. The transition probability is estimated by simple counting. We use the Gaussian mixture densities as emission probability distribution $Pr(x_t|s_t, w)$ in the states. The emission probability is defined as:

$$Pr(x_t|s_t, w) = \sum_{l=1}^{L(s_t, w)} Pr(x_t, l|s_t, w)$$
$$= \sum_{l=1}^{L(s_t, w)} Pr(l|s_t, w) \cdot Pr(x_t|s_t, w, l), \tag{5}$$

where $L(s_t, w)$ is the number of densities in each state and

$$Pr(x_t|s_t, w, l) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{l,s_t,w,d}^2}} \cdot \exp\left(-\frac{(x_{t,d} - \mu_{l,s_t,w,d})^2}{\sigma_{l,s_t,w,d}^2} \right). \tag{6}$$

In this work, the sum is approximated by the maximum, and the emission probability is defined as:

$$Pr(x_t|s_t, w) = \max_l Pr(x_t, l|s_t, w)$$
$$= \max_l Pr(l|s_t, w) \cdot Pr(x_t|s_t, w, l). \tag{7}$$

To estimate $Pr(x_t|s_t, w)$, we use the maximum likelihood estimation method for the parameters of the Gaussian distribution, i.e. the mean $\mu_{s_t,w,d}$ and the variances $\sigma_{s_t,w,d}$. Here, the covariance matrix is modeled to be diagonal, i.e. all off-diagonal elements are fixed at zero. The number of states for the HMM of each word is determined by the minimum sequence length of the training samples. Instead of a density-dependent estimation of the variances, we use pooling during

the training of the HMM, which means that we do not estimate variances for each density of the HMM, but instead we estimate one set of variances for all densities in the complete model (word-dependent pooling).

We use the Viterbi algorithm to find the maximizing state sequence s_1^T. In the Viterbi algorithm, we calculate the score of the observation feature vector x_t in the emission probability distribution $Pr(x_t|s_t, w)$ at each state s_t. Assuming the Gaussian function with diagonal covariances for $Pr(x_t|s_t, w)$, as described above, this score is calculated as:

$$- \log Pr(x_t|s_t, w) = \min_l \left\{ \frac{1}{2} \underbrace{\sum_{d=1}^{D} \frac{(x_{t,d} - \mu_{l,s_t,w,d})^2}{\sigma_{l,s_t,w,d}^2}}_{\text{distance}} - \log Pr(l|s_t, w) + \right.$$

$$\left. \frac{1}{2} \sum_{d=1}^{D} \log(2\pi\sigma_{l,s_t,w,d}^2) \right\}. \qquad (8)$$

In this work, the feature vector x_t is a down-scaled image at time t with a dimensionality of $D = 143$. Therefore, the sum $\sum_{d=1}^{D}(x_{t,d} - \mu_{l,s_t,w,d})^2 / \sigma_{l,s_t,w,d}^2$ is the distance between the observation image at time t and the mean image $\mu_{l,s_t,w}$ of the state s_t which is scaled by the variances $\sigma_{l,s_t,w,d}^2$. This scaled Euclidean distance can be replaced by other distance functions such as the tangent distance, which we will introduce in the following section.

The number of utterances in the database for each word is not large enough to separate them into training and test sets, for example some words of the database occur only twice. Therefore, we employ the leaving one out method for training and classification, i.e. we test the classifier on each sample in turn while training on the remaining 482 samples. The percentage of the misclassified utterances is the error rate of the system.

5   Tangent Distance

In this section, we give an overview of the distance measure invariant to affine transformations called *tangent distance*, which was first introduced in [9]. The incorporation into a statistical system was presented in [6]. An invariant distance measure ideally takes into account transformations of the patterns, yielding small values for patterns which mostly differ by a transformation that does not change class-membership.

Let $x_t \in \mathbb{R}^D$ be a pattern, and $x_t(\alpha)$ denote a transformation of x_t that depends on a parameter L-tuple $\alpha \in \mathbb{R}^L$, where we assume that this transformation does not affect class membership (for small α). The set of all transformed patterns is now a manifold $\mathcal{M}_{x_t} = \left\{ x_t(\alpha) : \alpha \in \mathbb{R}^L \right\} \subset \mathbb{R}^D$ in pattern space. The distance between two patterns can then be defined as the minimum distance between the manifold $\mathcal{M}_{x_t}$ of the pattern x_t and the manifold $\mathcal{M}_\mu$ of a class specific prototype pattern μ. This manifold distance is truly invariant with respect to the regarded transformations. However, the distance calculation between manifolds is a hard non-linear optimization problem in general. The

manifolds can be approximated by a *tangent subspace* $\widehat{\mathcal{M}}$. The *tangent vectors* $x_{t,l}$ that span the subspace are the partial derivatives of $x_t(\alpha)$ with respect to the parameters α_l $(l = 1, \ldots, L)$, i.e. $x_{t,l} = \partial x_t(\alpha)/\partial \alpha_l$. Thus, the transformation $x_t(\alpha)$ can be approximated using a Taylor expansion at $\alpha = 0$:

$$x_t(\alpha) = x_t(0) + \sum_{l=1}^{L} \alpha_l x_{t,l} + \sum_{l=1}^{L} \mathcal{O}(\alpha_l^2) \tag{9}$$

The set of points consisting of the linear combinations of the tangent vectors $x_{t,l}$ added to x_t forms the tangent subspace $\widehat{\mathcal{M}}_{x_t}$, a first-order approximation of $\mathcal{M}_{x_t}$:

$$\widehat{\mathcal{M}}_{x_t} = \left\{ x_t + \sum_{l=1}^{L} \alpha_l x_{t,l} \; : \; \alpha \in \mathbb{R}^L \right\} \subset \mathbb{R}^D \tag{10}$$

Using the linear approximation $\widehat{\mathcal{M}}_{x_t}$ has the advantage that distance calculations are equivalent to the solution of linear least square problems, or equivalently, projections into subspaces, which are computationally inexpensive operations. The approximation is valid for small values of α, which nevertheless is sufficient in many applications, as Fig. 4 shows for example of an image frame of BOSTON50 dataset. These examples illustrate the advantage of tangent distance over other distance measures, as the depicted patterns all lie in the same subspace and can therefore be represented by one prototype and the corresponding tangent vectors. The tangent distance between the original image and any of the transformations is therefore zero, while the Euclidean distance is significantly greater than zero. Using the squared Euclidean norm, the tangent distance is defined as:

$$d(x_t, \mu) = \min_{\alpha, \beta \in \mathbb{R}^L} \left\{ \left\| (x_t + \sum_{l=1}^{L} \alpha_l x_{t,l}) - (\mu + \sum_{l=1}^{L} \beta_l \mu_l) \right\|^2 \right\} \tag{11}$$

This distance measure is also known as two-sided tangent distance. To reduce the effort for determining $d(x_t, \mu)$, it may be convenient to restrict the tangent subspaces to the derivatives of the reference or the observation. The resulting distance measure is then called one-sided tangent distance. In this work, we replaced the Euclidean distance with the one-sided tangent distance using the derivatives of the mean image μ_{s_t} in state s_t.
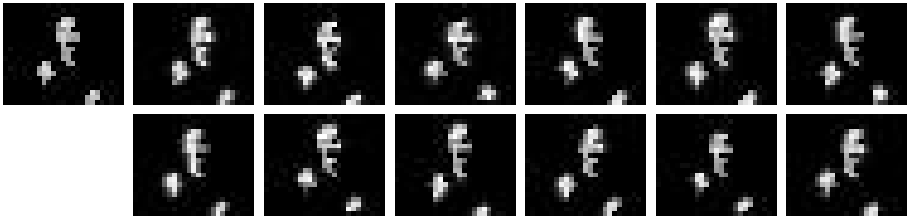
Fig. 4. Example of first-order approximation of affine transformations (Left to right: original image, ± horizontal translation, ± vertical translation, ±axis deformation, ± diagonal deformation, ± scale, ± rotation)

6   Clustering

Due to the high variability of utterances for each word in the database, we consider different pronunciations for utterances of each word. Note that this approach involves a tradeoff; while we may be able to better model the different pronunciations when we use separate HMMs, we are left with fewer data to estimate the HMMs from. We employ and compare three methods of clustering to determine the partitioning into clusters.

Manual Clustering. We observed that there are large visual differences between the utterances of each word, and that they are visually distinguishable. Thus, we are able to label the utterances of different pronunciations for each word as a baseline. We separated the 483 utterances of the BOSTON50 database to 83 pronunciations for the 50 words. The results obtained using this method serve as a lower bound for the automatic methods described in the following because we cannot hope to obtain a better cluster structure. Obviously, for any larger task it will not be feasible to perform a manual labelling. Interestingly, as the experimental results show, the automatic methods can yield error rates that are close to the ones obtained with manually selected labels.

k-means Clustering. One basic but very popular clustering approach is the k-means clustering method. In this method the number of clusters is assumed to be known beforehand and equal to k. We choose one utterance of each of the clusters that were labeled manually as a seed in the initialization. The algorithm continues by adding other utterances to the cluster.

In this algorithm for all words of the database: after initializing k (number of the clusters) and calculating the μ_i as the mean of a the Gaussian function made by utterances of each cluster, all samples would be classified to the nearest cluster. This would be repeated until no change happens in clusters.

LBG-Clustering. The k-means clustering still uses some manually extracted information, i.e. the number of clusters and the initializing seeds of the clusters. We employ the LBG-clustering algorithm proposed by [10] to overcome this constraint and obtain a fully automatic clustering algorithm. This method is described as follows: We perform the clustering for all words of the database as it is shown in Figure 5. First, we assume that all utterances belong to one cluster or particular pronunciation and create an HMM with all utterances existing for a word. If the criterion for dividing a cluster is met, we divide this HMM into two
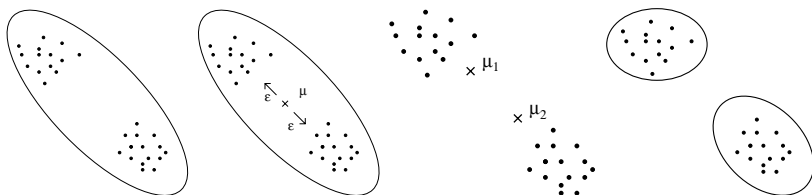
Fig. 5. The LBG-clustering

new cluster centers by adding or subtracting a small value to all means of the states in the model. Then we calculate the similarity between all possible pairs of cluster centers for the word and merge them if the criterion for merging is met. We continue to divide and merge the clusters until no change in the cluster assignment occurs.

The criterion function is defined to calculate the dispersion or scattering of the utterances in a cluster. We use the mean squared distance of the utterances to the mean model as a measure of scatter and normalize that value to the range $[0, 1]$. We consider a threshold value for this criterion function to control the coarseness of the clustering.

Nearest Neighbor Classifier. Nearest neighbor classification is a special case in modelling of the different pronunciations. In nearest neighbor classification the number of pronunciations is considered to be equal to the number of the training utterances for each word. Using each training utterance in the database, we create an HMM. According to the leaving one out method used in this work we separate an utterance as a test utterance from the database. This unknown utterance is classified as belonging to the same class as the most similar or nearest utterance in the training set of the database. This process is repeated for all utterances in the database.

7 Experimental Results

The experiments have been started by employing an HMM for each word of the BOSTON50 database resulting in an error rate of 28.4% with Euclidean distance. We repeated the experiment using the different proposed clustering methods and the tangent distance.

The results are summarized in Table 1. The results show that, in all experiments, tangent distance improves the error rate of the classifiers by between 2 and 10 percent relative. Furthermore, employing clustering methods and the nearest neighbor classifier yields a lower error rate than obtained without considering different pronunciations. The threshold value used in LBG-clustering is a normalized value. When the threshold value is set to 1, no clustering occurs, and when it is set to 0 each utterance will form a separate cluster and the classifier converges to the nearest neighbor classifier. The error rate of the classifier using LBG-clustering with respect to the threshold value is shown in Fig. 6. We

Table 1. Error rates [%] of the HMM classifier with different distances and clusterings

	Euclidean Distance	Tangent Distance
No Clustering	28.4	27.7
Manual Partitioning	22.8	**20.5**
k-means Clustering	23.8	21.3
LBG Clustering	23.2	21.5
Nearest Neighbor	23.6	22.2
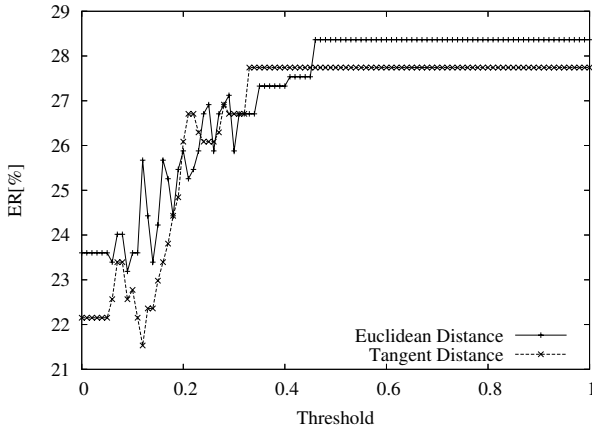
Fig. 6. Error rate of the system with respect to the threshold of clustering

can observe that, with a threshold value of 1, no clustering happens and the error rate is equal to the error rate of the classifier without any pronunciation modeling. When decreasing the threshold value, the error rate is reduced and we can achieve the best error rate of 23.2% and 21.5% using the Euclidean distance and the tangent distance, respectively. The fluctuations we can observe in the diagram for threshold values between 0 and 0.4 lead us to the conclusion that the determination of the best threshold value is not very reliable. Nevertheless, we can observe that there is a strong trend of reducing error rates for smaller threshold values. This leads us to consider the nearest neighbor classifier, which corresponds to the threshold value zero and achieves error rates of 23.6% and 22.2% with the Euclidean distance and the tangent distance, respectively. Because these values are only slightly less than the best –but unstable– result for LBG clustering, this approach should be considered for tasks with a large variability of utterances.

The best error rate of 20.5% is achieved using manual clustering and tangent distance but the results achieved using other clustering methods will be preferable for large databases because they do not involve human labeling of video sequences. The best pronunciation clustering method without human intervention is the hierarchical LBG-clustering with tangent distance and an error rate of 21.5%, which is an improvement of over 22 percent relative.

In the experiments reported above, mixture densities with a maximum number of five densities are used in each state. We have repeated the experiments employing single density and mixture densities, consisting of more densities, in the states of the HMMs. Table 2 shows the results of the experiments employing the tangent distance and different clustering methods. The results show that using a higher number of densities within a mixture density improves the accuracy of the system. In other words, the mixture densities can model the variability of the utterances even without employing the clustering methods. The error rate of the system without any clustering method is 22.8%. In most experiments, the better results are achieved when mixture densities are used in the states. When

Table 2. Error rates [%] of the HMM classifier employing single and mixture densities

	Single Density	Mixture Density
No Clustering	47.4	22.8
Manual Partitioning	35.4	21.9
k-means Clustering	33.1	21.1
LBG Clustering	21.7	22.1

mixture densities are used, the influence of different clustering methods on the error rate of the system is much less than single density experiments.

About half of the remaining errors are due to visual singletons in the dataset, which cannot be classified correctly using the leaving one out approach. This means that one word was uttered in a way that is visually not similar to any of the remaining utterances of that word. For example, all but one of the signs for POSS show a movement of the right hand from the shoulder towards the right side of the signer, while the remaining one shows a movement that is directed towards the center of the body of the signer. This utterance thus cannot be classified correctly without further training material that shows the same movement. This is one of the drawbacks of the small amount of training data available.

A direct comparison to results of other research groups is unfortunately not possible here, because there are no results published on publicly available data so far, and research groups working on sign language or gesture recognition usually use databases that were created within the group. We hope that other groups will produce results for comparison on the BOSTON50 database in the future.

8   Conclusion

In this paper we introduced an appearance-based sign language recognition system. According to our results, considering different pronunciations for sign language words improves the accuracy of the system.

Due to the modeling of different pronunciations of each word in the database, we employed three kinds of the clustering methods; manual clustering, k-means clustering and hierarchical LBG-clustering. These methods can be chosen according to the size of the database in different applications.

Although manual clustering gives more accuracy, it needs manually extracted information and can therefore only be employed for small sets of data. The k-means clustering needs less initial information and only needs to be initialized with the number of clusters and manually selected seed utterances, so this method is also suitable for medium size databases. In contrast, the LBG-clustering method partitions the data automatically and is preferable for large databases where extracting labels manually is unfeasible. According to the results of the experiments on the BOSTON50 database, LBG-clustering leads us to use the nearest neighbor classifier that performs surprisingly well. In all experiments, the tangent distance was compared to the Euclidean distance within the Gaussian emission densities. Using the tangent distance that models small global affine transformations of the images improves the accuracy of the classifier significantly.

References

1. Y. Nam and K. Wohn. Recognition of Space-Time Hand-Gestures Using Hidden Markov Model. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pp. 51–58, Hong Kong, July 1996.
2. B. Bauer, H. Hienz, and K.F. Kraiss. Video-Based Continuous Sign Language Recognition Using Statistical Methods. In *Proceedings of the International Conference on Pattern Recognition*, pp. 463–466, Barcelona, Spain, September 2000.
3. T. Starner, J. Weaver, and A. Pentland. Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 20(12):1371–1375, December 1998.
4. C. Vogler and D. Metaxas. Adapting Hidden Markov Models for ASL Recognition by Using Three-dimensional Computer Vision Methods. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pp. 156–161. Orlando, FL, October 1997.
5. M. Zahedi, D. Keysers, and H. Ney. Appearance-based Recognition of Words in American Sign Language. *2nd Iberian Conference on Pattern Recognition and Image Analysis*, Volume LNCS 3522 of Lecture Notes in Pattern Recognition and Image Analysis, pp. 511–519, Estoril, Portugal, June 2005.
6. D. Keysers, W. Macherey, and H. Ney. Adaptation in Statistical Pattern Recognition Using Tangent Vectors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):269–274, February 2004.
7. C. Neidle, J. Kegl, D. MacLaughlin, B. Bahan, and R.G. Lee. *The Syntax of American Sign Language: Functional Categories and Hierarchical Structure*. MIT Press, Cambridge, MA, 2000.
8. L.R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 77(2):267–296, February 1989.
9. P. Simard, Y. Le Cun, and J. Denker. Efficient Pattern Recognition Using a New Transformation Distance. In *Advances in Neural Information Processing Systems 5*, pp. 50–58, Morgan Kaufmann, 1993.
10. Y. Linde, A. Buzo, and R. Gray. An Algorithm for Vector Quantization Design. *IEEE Trans. on Communications*, Vol. 28, pp. 84–95, January 1980.

Appendix: Visual Lexicon Data

The BOSTON50 database consists of 50 sign language words that are listed with the number of occurrences here:

IX_i (37), BUY (31), WHO (25), GIVE (24), WHAT (24), BOOK (23), FUTURE (21), CAN (19), CAR (19), GO (19), VISIT (18), LOVE (16), ARRIVE (15), HOUSE (12), IX_i"far" (12), POSS (12), SOMETHING/ONE (12), YESTERDAY (12), SHOULD (10), IX-1p (8), WOMAN (8), BOX (7), FINISH (7), NEW (7), NOT (7), HAVE (6), LIKE (6), BLAME (6), BREAK-DOWN (5), PREFER (5), READ (4),COAT (3), CORN (3), LEAVE (3), MAN (3), PEOPLE (3),THINK (3), VEGETABLE (3) VIDEOTAPE (3), BROTHER (2), CANDY (2), FRIEND (2), GROUP (2), HOMEWORK (2), KNOW (2),LEG (2), MOVIE (2), STUDENT (2), TOY (2), WRITE (2).

Visual Sign Language Recognition Based on HMMs and Auto-regressive HMMs

Xiaolin Yang[1], Feng Jiang[1], Han Liu[2], Hongxun Yao[1], Wen Gao[1,3], and Chunli Wang[3]

[1] Department of Computer Science, Harbin Institute of Technology, Harbin, China,15001
{yangxl, fjiang, yhx}@vilab.hit.edu.cn
[2] Department of Computer Science, Univerity of Illinois at Urbana-Champaign,
Champaign, IL, USA, 61820
hanliu2@uiuc.edu
[3] Institute of Computing Technology, Chinese Academy of Science, Beijing, China, 100080
{wgao, clwang}@ict.ac.cn

Abstract. A sign language recognition system based on Hidden Markov Models(HMMs) and Auto-regressive Hidden Markov Models(ARHMMs) has been proposed in this paper. ARHMMs fully consider the observation relationship and are helpful to discriminate signs which don't have obvious state transitions while similar in motion trajectory. ARHMM which models the observation by mixture conditional linear Gaussian is proposed for sign language recognition. The corresponding training and recognition algorithms for ARHMM are also developed. A hybrid structure to combine ARHMMs with HMMs based on the trick of using an ambiguous word set is presented and the advantages of both models are revealed in such a frame work.

Keywords: Computer Vision, Sign Language Recognition, HMM, Auto-regressive HMM.

1 Introduction

Visual sign language recognition aroused many researcher's interests nowadays. The successful application of HMMs to speech recognition brought the ideas of using it in sign language recognition. Starner [1] presented two video-based systems for real-time recognizing sentence-level continuous ASL. Some extension of HMMs was also applied to sign language recognition. Tatsuya Ishihara [2] et al provides a method to recognize gestures using auto-regressive coefficients of features.Traditional HMMs only consider the relationship between every state, while the information between observations has been lost. To solve this problem we present a novel method to incorporate auto-regressive HMMs in our original system based on traditional HMMs.
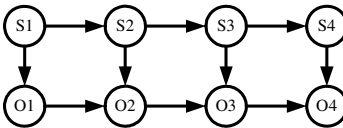
Fig. 1. Auto-regressive HMMs

2 Auto-regressive HMMs and Parameter Estimation

The regular HMMs assumes that the observations are conditionally independent given the hidden state is quite strong, and can be relaxed at little extra cost. This model reduces the effect of the N_t "bottleneck", by allowing O_t to be predicted by O_{t-1} as well as N_t, this results in models with higher likelihood. Figure 1 illustrates this kind of ARHMM that we use in this paper. Mixture conditional linear Gaussian function is implemented to model the consecutive signal. That is,

$$b_j(O) = \sum_{m=1}^{M} c_{jm} N[B_{jm}O_{t-1} + \mu_{jm}, \Sigma_{jm}, O_t], 1 \le j \le N ,$$ (1)

where c_{jm} is the mixture coefficient. $\sum_{m=1}^{M} c_{jm} = 1$, $c_{jm} > 0$ and $1 \le j \le N$,

$1 \le m \le M$, M is the number of mixture terms. The estimation of regression matrix B_i [3] for the observational density function is

$$B_i = (\sum_l \omega_m^i \langle y_m x_m' \rangle_i)(\sum_l \omega_m^i \langle x_m x_m' \rangle_i)^{-1}$$ (2)

Where $\omega_m^i = P(S = i \mid D_m)$ $D = \{D_1, \cdots, D_M\}$ is the training set. In case of mixture conditional linear Gaussian, we can get the estimation of B_{jm} as:

$$\overline{B}_{jm} = \sum_{k=1}^{K} \sum_{t=2}^{T_k} \sum_{j=1}^{N} \gamma_t^k(j,m) \langle o_t, o_{t-1} \rangle \bigg/ \sum_{k=1}^{K} \sum_{t=2}^{T_k} \sum_{j=1}^{N} \gamma_t^k(j,m) \langle o_{t-1}, o_{t-1} \rangle$$ (3)

$\gamma_t^k(j,m)$ is the conditional probability density of the data which comes from the t th frame and is at state j and is the m th term in the mixture Gaussian function. The reestimate formulations of $\pi_i, a_{ij}, c_{jm}, \mu_m, \Sigma_m$ could be calculated as that of the standard HMMs.

3 Sign Language Recognition Systems

As shown in Figures 2 and 3, this system measures hand gestures using input devices USB PC color video camera. The feature extraction process can be referred to [4]. Both HMMs and ARHMMs are applied under the trick of ambiguous word set. We here adopt the hybrid structure in our system because the lack of training data is a main problem in applying ARHMM to sign language recognition. The basic idea is illustrated in Figure 3. The recognition result list is generated from the ambiguous word set in which words accurately recognized by HMMs correspond only one word otherwise correspond several words in the ambiguous word set.

Both HMMs and auto-regressive HMMs are used in our system. As shown in figure 2, the ARHMMs can be viewed as a refiner classifier, which could also be called a "tuner". After the video data being processed and the features extracted, the feature data was input to the HMMs. Figure 3 gives the detail of double layer recognition process. For every HMM, we use a large amount of samples to test the models. So a
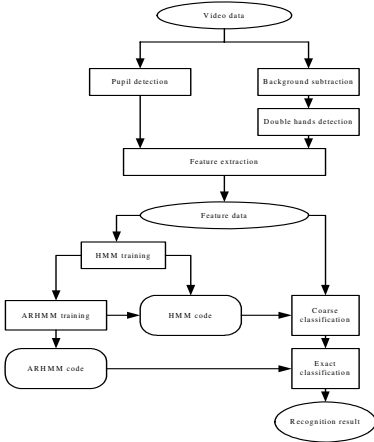
Fig. 2. System overview
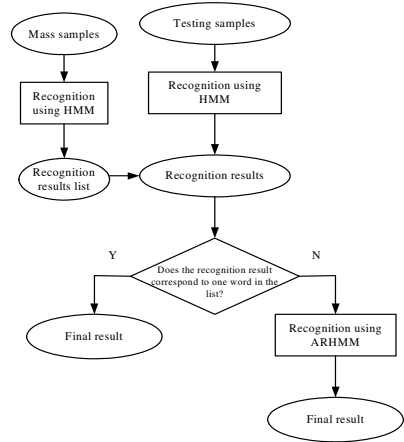
Fig. 3. Double layer recognition scheme

Table 1. Recognition result of HMM

number of word to be recognized	number of wrongly recognized word	recognition accuracy
439	35	92.0%

Table 2. Recognition result of HMM and ARHMM

number of word to be recognized	number of wrongly recognized word	recognition accuracy
439	15	96.6%

(a)   (b)
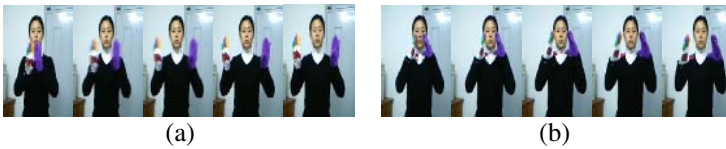
Fig. 4. Two signs which can't be discriminated by regular HMM (a) inaugurate (b) propagandize

certain model may correspond to one word of the right recognition result or several words which are wrongly recognized as that word. This process is called a coarse classification. After initializing B, we train the ARHMMs, Section 3 gives the details of parameter estimation.

The process of recognition is to choose a model which describes the observation signal the best from the candidate models set. Since our system is double-layer with HMMs and ARHMMs. Once there is an unknown sign waiting to be recognized, it should be first classified by HMMs. If it is recognized as a word which does not appear in the ambiguous word set, it would not be reclassified by ARHMM. Otherwise, if it is recognized as a word whose model corresponds to several words, the word will be classified further by ARHMMs. Therefore, both models contribute to the recognition accuracy of our system. The advantages of both models have been fully considered.

4   Experiments and Results

We collect the signs in the 439-sign lexicon, each 5 times for every signs i.e. 2195 signs are collected, where 4 times data are used to build the training set and the remaining 1 time data to build the testing set.

The word recognition accuracy of the lexicon signs based on HMMs is shown in Table 1. After incorporating ARHMMs to our original system, we get the recognition result in Table 2.We can see that the word recognition accuracy improves 4.6% after incorporating ARHMMs to the system.

It shows that the regression matrix of ARHMMs is good at modeling linear motion trajectory. For example, in Figure 4 "inaugurate" and "propagandize" are double-hand words with hands moving aside. ARHMMs based on conditional linear Gaussian can describe this observation well. The auto-regressive matrix gives a better description of the hand motion than regular HMMs. But with no constraints on the lighting condition and background, the features of some frames can not be detected or assumed as abnormal values directly because we can "see" these features in two-dimensional view. These factors are the main reasons that our ARHMMs based on conditional linear Gaussian has worse performance than regular HMMs in modeling some signs. Our two layers' classifier complements the deficiency of both models. So ARHMMs helps to recognize confusing words and may be a good method for large vocabulary sign language recognition.

5   Conclusions

In this paper, we present a method to model temporal signals ARHMM and a hybrid structure to combine both ARHMMs and HMMs to recognize isolated sign language words. The experimental result shows that the ARHMMs can greatly improve the whole recognition rate for its good ability to model the linear movement. Future work should add 3-D information to features to improve the performance of ARHMMs; Other observation representation should also be explored for a better description of observational signals.

References

[1]   T. Starner, J. Weaver, and A. Pentland, "Real-time American sign language recognition using desk and wearable computer based video", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12): 1371-1375, December 1998.

[2]   Ishihara, T. Otsu, "Gesture recognition using auto-regressive coefficients of higher-order local auto-correlation features", sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004.

[3]   K.P. Murphy, "Dynamic Bayesian Networks: Representation, Inference and Learning", pp.185-187

[4]   Liangguo Zhang, Yiqiang Chen, Gaolin Fang, Xilin Chen, Wen Gao. A Vision-Based Sign Language Recognition System Using Tied-Mixture Density HMM. ACM 6th International Conference on Multimodal Interfaces (ICMI'04), State College, Pennsylvania, USA, Oct.14-15, 2004, pp. 198-204.

A Comparison Between Etymon- and Word-Based Chinese Sign Language Recognition Systems*

Chunli Wang[1,2], Xilin Chen[1], and Wen Gao[1]

[1] Institute of Computing Technology, Chinese Academy of Science, 100080, Beijing, China
[2] Department of Computer Science and Engineering,
School of Electronic and Information Engineering,
Dalian University of Technology, 116023, Dalian, China
{clwang, xlchen, wgao}@jdl.ac.cn

Abstract. Hitherto, one major challenge to sign language recognition is how to develop approaches that scale well with increasing vocabulary size. In large vocabulary speech recognition realm, it is effective to use phonemes instead of words as the basic units. This idea can be used in large vocabulary Sign Language recognition, too. In this paper, Etyma are defined to be the smallest unit in a sign language, that is, a unit that has some meaning and distinguishes one sign from the others. They can be seen as phonemes in Sign Language. Two approaches to large vocabulary Chinese Sign Language recognition are discussed in this paper. One uses etyma and the other uses whole signs as the basic units. Two CyberGloves and a Pohelmus 3-D tracker with three receivers positioned on the wrist of CyberGlove and the back are used as input device. Etymon- and word- based recognition systems are introduced, which are designed to recognize 2439 etyma and 5100 signs. And then the experimental results of these two systems are given and analyzed.

1 Introduction

Sign language, as a kind of structured gesture, is one of the most natural means of exchanging information for most deaf people. It is a kind of visual language via hand and arm movements accompanying facial expressions and lip motions. The aim of sign language recognition is to provide an efficient and accurate mechanism to translate sign language into text or speech.

Attempts to automatically recognize sign language began to appear at the end of 80's. T.Starner [1] achieved a correct rate of 91.3% for 40 signs based on the image. By imposing a strict grammar on this system, the accuracy rates in excess of 99% were possible with real-time performance. Fels and Hinton [2][3] developed a system using a VPL DataGlove Mark II with a Polhemus tracker as input devices. Neural network was employed for classifying hand gestures. R.H.Liang and M. Ouhyoung[4] used HMM for continuous recognition of Tainwan Sign language with a vocabulary between[71 and 250 signs by using Dataglove as input devices. C. Wang[5] realized a continuous Chinese Sign Language (CSL) recognition system with a vocabulary of 5100 signs. C. Vogler and D. Metaxas[6] described an approach to continuous,

---

whole-sentence ASL recognition, in which phonemes instead of whole signs were used as the basic units. They experimented with 22 words and achieved similar recognition rates with phoneme-based and word-based approaches.

From the review of the previous researches above mentioned, we know that most researches on continuous sign language recognition were done on small test vocabulary. The major challenge to sign language recognition is how to develop approaches that scale well with increasing vocabulary size. In speech recognition, using phonemes as basic unit assuredly is an effective solution to large vocabulary system. Is this idea also useful in large vocabulary sign language recognition?

In this paper, we discuss two approaches to large vocabulary Chinese Sign Language (CSL) recognition. One uses etyma and the other uses whole signs as the basic units. The results of these two approaches are compared.

2   Etymon-Based System

Two CyberGloves and a Pohelmus 3-D tracker with three receivers positioned on the wrist of CyberGlove and the back are used as input device. The raw gesture data include hand postures, positions and orientations. A sign is a sequence of frames. A frame of the raw gesture data, which in our system are obtained from 36 sensors on two datagloves, and three receivers mounted on the datagloves and the waist, are formed as 48-dimensional vector. A dynamic range concept is employed in our system for satisfying the requirement of using a tiny scale of data. The dynamic range of each element is different, and each element value is normalized to ensure its dynamic range 0-1.

Here, one etymon is defined to be the smallest unit in a sign language, that is, a unit that has some meaning and distinguishes one sign from another. For example, "Teacher" is composed by two etyma, which are shown in Fig. 1. The Bopomofo are considered as etyma, which can facilitate the CSL recognition when finger-alphabet is used accompanying with gestures. Unlike the etyma in spoken language, no explicit definition of the etymon exits in the CSL linguistics. Based on extensive and thorough analysis of 5100 signs in CSL, we find all the units that form all the signs in the CSL dictionary. Finally, about 2400 etyma are explicitly defined for CSL.

The sign data collected by the gesture-input devices is fed into the feature extraction module, and then the feature vectors are input into the training module, in which a model is built for each etymon. The signs are encoded based on the etyma, and the Etymon-sequences of signs are stored in a codebook, based on which the tree-structured network and forward index tables are built to reduce the search range [7].
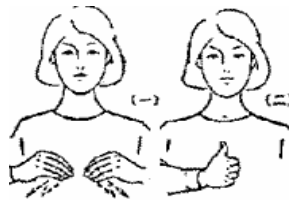
Fig. 1. Two Etyma in the word "Teacher"

The language model that is used in our system is Bi-gram model. The decoder controls the search for the most likely priority of sign appearance in a sign sequence. Then the sign sequence is exported from the decoder.

3   Experimental Results Comparison

In order to compare these two methods, some experiments are carried out. 5100 signs in CSL dictionary are used as evaluation vocabulary. Each sign was performed five times, four times are used for training and one for testing. 2439 etyma are defined for CSL. Each Etymon was performed five times for training. The number of states in HMMs is 3. 5100 signs are coded with these etyma automatically. Experiments are done based on word and etyma respectively.

One sign consists of one or more etyma. Each sign is a string of etyma. Therefore, there is movement epenthesis between two etyma, which will affect the recognition results. For the system based on etyma, recognizing a sign is similar to recognizing a sentence based on signs, so the recognition rate of isolated signs will decline.

There are about 2400 basic units and 5100 signs in Chinese sign language. The numbers of candidates in these two approaches are equivalent because of the using of Viterbi-beam algorithm. Therefore, the time of loop in the system based on etyma is half of that in the system based on signs when selecting the candidates. In the following two parts, the times of loop are the same. Besides, the algorithm based on etyma is more complex, and it takes more time to decode. The approach based on etyma does not gain better effect on the aspect of improving the speed.

The 5100 signs and 2439 etyma are analyzed and it is discovered that many etyma have low appearing frequencies in all words, namely they only appear in one or two words. For example, the etymon "electric car" is a word by itself, and does not appear in any other words. If throwing off the etyma that appear less than twice in 5100 words, there are 723 etyma left. These etyma compose 3048 words. If deleting the etyma that appear less than three times in 5100 words, there are 581etyma and 2650 words left. Experiments are done with these three vocabularies, and the results are shown in Fig. 2.

(1) Time   (2) Accuracy
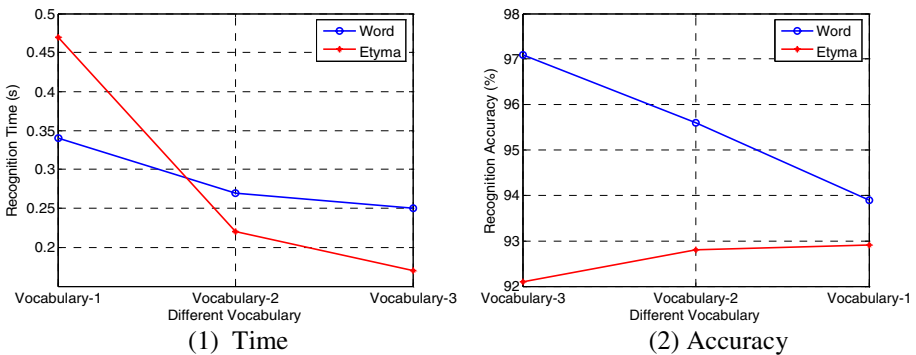
Fig. 2. The comparison of the performances of two systems based on different size of vocabulary. Vocabulary-1: 2439 Etyma and 5100 signs; Vocabulary-2: 723 Etyma and 3048 signs; Vocabulary-3: 581 Etyma and 2650 signs.

For the 5100 isolated words, the recognition rate of the approach based on etyma is lower than that of the approach based on whole signs. The recognition time based on etyma is longer than that based on words. But when the number of etyma with low appearing frequencies in all words decreases, the performances based on etyma are improved. In the case that the size of the vocabulary is large and the number of words is four times more than the number of etyma, the approach based on etyma is the proper selection.

4   Conclusions and Future Work

In speech recognition, using phonemes as basic unit assuredly is an effective solution to large vocabulary system. But is it also useful in sign language recognition? In this paper, two approaches to large vocabulary CSL recognition are introduced. Experimental results of these two approaches are compared.

There are, however, many problems that still need to be resolved. The impact of the movement between two signs is not eliminated, yet. Context-dependent Etymon (TRIPHONE) models will be built to solve this problem. But because the number of etyma is much more than that of spoken language, the effect won't be obviously. How to reduce the number of the etyma is the key.

References

1. Starner T., Weaver J., Pentland A.: Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video. IEEE PAMI, Vol 20, Issue 12, Dec 1998, pages 1371-1375.
2. S.S.Fels, G.Hinton.: GloveTalk:A neural network interface between a DataDlove and a speech synthesizer. IEEE Transactions on Neural Networks, 4(1993):2-8.
3. S.Sidney Fels.: Glove –TalkII: Mapping hand gestures to speech using neural networks-An approach to building adaptive interfaces. PhD thesis, Computer Science Department, University of Torono, 1994.
4. R.-H.Liang, M.Ouhyoung.: A real-time continuous gesture recognition system for sign language. In Proceeding of the Third International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pages 558-565.
5. Chunli Wang, Wen Gao.: A Real-time Large Vocabulary Recognition Continuous System for Chinese Sign Language. Advances in Multimedia Information Processing-PCM 2001, Beijing, China, October 2001, 150-157.
6. ChristianVogler, Dimitris Metaxas.: Toward scalability in ASL Recognition: Breaking Down Signs into Phonemes. In Proceedings of Gesture Workshop, Gif-sur-Yvette, France, 1999, pages 400-404.
7. Chunli Wang, Wen Gao, Shiguang Shan.: An approach based on phonemes to large vocabulary Chinese sign language recognition. Proceeding of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (FG'02), Washington, USA, 2002, 411-416.

Real-Time Acrobatic Gesture Analysis

Ryan Cassel[1], Christophe Collet[2], and Rachid Gherbi[1]

[1] LIMSI-CNRS, Université de Paris XI, BP 133, 91403 Orsay cedex, France
[2] IRIT, Université Paul Sabatier, 31062 Toulouse cedex, France
{cassel, gherbi}@limsi.fr, collet@irit.fr

Abstract. Gesture and motion analysis is a highly needed process in the athletics field. This is especially true for sports dealing with acrobatics, because acrobatics mix complex spatial rotations over multiple axes and may be combined with various postures. This paper presents a new vision-based system focused on the analysis of acrobatic gestures of several sports. Instead of classical systems requiring modelizing human bodies, our system is based on the modelling and characterization of acrobatic movements. To show the robustness of the system, it was successively tested first on movements from trampoline, and also in other sports (gymnastics, diving, etc.). Within the system, the gestures analysis is mainly carried out by using global measurements, extracted from recorded movies or live video.

1 Introduction

In computer vision systems, techniques of motion analysis are increasingly robust and beginning to have an impact beyond laboratories. Such systems could be useful in order to evaluate the performance of gestures in many applications. We separate communication gestures and sports gestures. These two fields of application use similar algorithms. In the context of communication gesture analysis many studies are devoted to sign language recognition. This task is well known to be hard to perform both in the hand/body/face tracking and recognition processes and in the analysis and recognition of the signs [3][9]. Other gestural studies deal more extensively with topics such as recognition of human activities [1]. The athletics sector is in strong demand for movement analysis. The advent of video techniques in this field already assists users because the video doesn't disturb sport gestures (it is non intrusive techniques). However, there are only few tools allowing automatic analysis in real time, while this type of analysis is necessary for many live sport performances. This paper addresses some of the representative publications on automatic systems of sporting gesture analysis. Yamamoto [4] presents a qualitative study about sporting movement (skiing). The aim of his study is to discriminate movements performed by people classified from novice to experts. Another work by Gopal Pingali [8] shows a system dealing with real time tracking and analysis of a tennis ball trajectory. This system was used for the tennis US Open 2000. The interest of such systems is obvious for many sports. But building them is a real challenge. The sporting context is very constraining from the environmental and gestural points of

view. Our study addresses acrobatic gestures. These gestures present a great diversity of complex movements. Each sport based on acrobatics requires a refined analysis in order to improve or judge the gesture quality, and this type of analysis is not commonly available. Sport organizations look forward to the development of such systems, and they agree with the idea that these systems could be a useful complement for training and a helpful tool for judges during competitions. The use of video techniques (non intrusive techniques) derives from specifications set by practice conditions. Thus, the athlete will be never constrained by the system, by contrast with systems involving active sensors for example. Many studies dealing with gesture analysis use sensors placed on the body to extract placement coordinates efficiently. Such systems are known as intrusive, they disturb the human movement making them less natural and they require a complex installation. Acrobatics make the use of sensors problematic. Capture devices such as sensors are expensive, invasive and constraining. However they are very precise compared to image processing. This paper addresses the development and assessment of an analysis system focused on acrobatic gestures. The system uses a fixed monocular passive sensor. The adopted approach is based on movement characterization initially used in trampoline competition. This movement model was improved and is briefly presented in section 2. The architecture system and its corresponding algorithms are described in section 3. Then, section 4 show results issued from evaluation of the system's robustness. Section 5 presents the use of the system in real situations of gymnastic gestures. Finally, a conclusion and some prospective comments are developed in the two last parts.

2   Model of Movement

The system design is based on a model of movement established and used by trampolinists (for more details refer to the *Code of Points* of the Fédération Internationale de Gymnastique [11]). It is based on chronological and axial movement decomposition. The model divides the movement into three parts. The first part relates to the quantity of transversal rotations of the body. The second relates to longitudinal rotations of the body distributed by the quantity of transversal rotations (see example below). The third relates to the body posture. This model of movement leads to a numerical notation described in more detail in [2]. This notation is built as follows:

$$s \; q \; v_1 \dots v_n \; p$$

$$s = [b|f]$$

$$q = [0-9]^*$$

$$v_i = [0-9]^*$$

$$p = [o| < |/]$$

Tuck   Pike   Layout
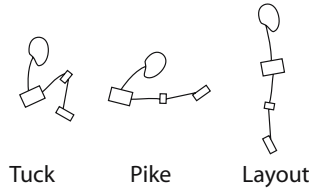
Fig. 1. Human body's positions in trampoline

where s indicates the direction of the figure (f, forward or b, backward), q describes the number of somersaults, by quarters, $v_1 \ldots v_n$ represent the distribution and quantity of half twists in each somersault and finally p describes the shape of the figure (tuck $= o$, pike $=<$ or layout $= /$ figure 1). Thus, $b\ 8\ 0\ 0\ o$ describes a double backward tuck somersault (720 of rotation) with no twist whereas $f\ 4\ 1\ /$ describes a forward layout somersault (360 of rotation) with a half twist.

This notation is valid for all acrobatic activities because it makes possible to identify any acrobatic movement. The recognition of each part of this notation informs us about the quality of realization. It is the basis of our analysis system. Section 5 shows some examples of use of such a system in gymnastics practice (trampoline).

3   System Architecture

The system comprises several connected modules in a hierarchical way. A lower layer level extracts relevant information (typically, pixels of the acrobat). A higher level layer transforms this information into interpretable data and analyzes this data. We present these various layers here.

3.1   Lower Level

The lower level layer extracts pixels of the acrobat. Our approach is to build a statistical model of background image and then to use image subtraction to emphasize the moving elements. We build an original method to eliminate noise, we called it *Block filtering*. And to optimize the acrobat extraction, we use Kalman filtering.

Background Model. In this paper, the term background refers to the pixels which are not moving. Thus, as far as the system is concerned, there is only one person in field of the camera. The gymnast is always moving while the background remains constant. However, in training or competition conditions the background is never fixed. Light variations and people passing behind the athlete make a background image vary. To adapt the background image variations, we use an adaptative generation of background. This generation is based on the luminance

mean of the N last images. The mean is calculated as $m_{(x,y)} = \frac{S_{(x,y)}}{N}$ where $S_{(x,y)}$ is the sum of pixel values in the location (x, y) and N is the number of the last frames collected. Subtraction between the current image and the background highlights fast parts moving. Many pixels are not belonging to the acrobat and are classified as noisy pixels. A filtering must be applied.

Block Filtering. drops noisy pixels. We include in noisy pixels, any moving pixel witch is not belonging to the gymnast. The acrobat is in the foreground of the camera recorder. The subtraction presented above leads to a binary image which contains pixels from the gymnast and pixels belonging to other moving objects. The perspective makes the person in the foreground bigger than other moving objects. Block filtering keeps only the biggest components of the binary image. The size of elements to be dropped is gauged by the size of blocks. To complete this filtering, the image is divided into blocks of size $(n \times m)$ (a grid of size $n \times m$ is applied on the binary image, it will be called *grid block*). Each block is marked as valid, invalid or adjacent. After computing all blocks, the system keeps only valid blocks and adjacent blocks. Invalid blocks are dropped. A block is marked as valid when the proportion of binary pixels is superior to a certain threshold. In the other case it is marked as invalid except if an adjacent block is valid (in a 8-connexity). In this case, the block is marked as adjacent. By defining the block size as larger than the noisy pixels elements' size (moving persons in the background for example), and by defining a threshold greater than the noise elements, the noisy pixels are dropped. Figure 2 shows an example of the algorithm. The upper left image corresponds to the original image. The upper right image corresponds to the result of the subtraction operation. It shows many pixels belonging to the noise created by elements moving in the background. The result of block filtering is given in the last image.

Let C be the mathematical expression for a block at pixel (x, y) on the grid block :

$$C(x,y) = \frac{\sum_{i=n.(x-1)}^{n.((x-1)+1)} \sum_{j=m.(y-1)}^{m.((y-1)+1)} I_{bin}(i,j)}{n.m}$$

where $I_{bin}(i, j)$ is the binary image at pixel (i, j). The criteria for selecting valid, invalid and adjacent block are defined as follow :

$$Block(x,y) = \begin{cases} \text{Valid,} & \text{if } C(x,y) \geq Th; \\ \text{Adjacent,} & \text{if } Block(x+i, y+j) = Valid \; \forall i \in [-1; 1], \forall j \in [-1; 1]; \\ \text{Invalid,} & \text{otherwise.} \end{cases}$$

where $Block(x, y)$ is a block at pixel (x, y) and Th the percentage of pixels wich should fill the block (fixed by user).

This filter is efficient when there is enough difference between foreground and background. However, a collective movement of the audience like a "holla" strongly disturb the algorithm. In this paper, the corpus used for this study does not include such situations.
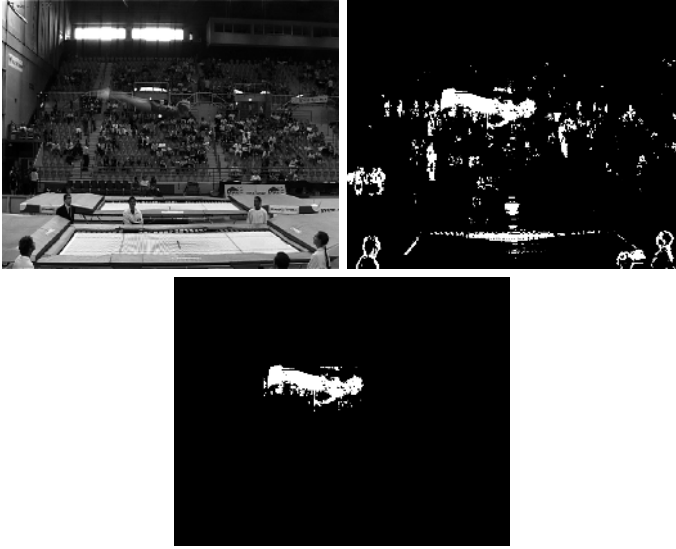
Fig. 2. Original image (top left), Binary image (top right) and Block Filtered image (bottom)

Kalman Filter. To improve time processing, it is important not to compute the entire images. Only the acrobat is relevant. Consequently, it is necessary to track the acrobat. Kalman filtering allows the system to predict and estimate displacement [5].

The region of interest of the image is reduced to a box around the gymnast (bounding box). All computing tasks are reduced to this bounding box. The Kalman filtering gives efficient prediction of the box's location. In addition to reducing the time processing, the bounding box focuses on the athlete. New elements around the bounding box do not come to disturb processing (for example, a person passing in the background). After an initialisation stage, the system is focused on the gymnast.

3.2   Motion Analysis (High Level)

Our characterization defines acrobatics with axial descriptions and with human body shape. Motion analysis part first extracts body axis and then analyses the body shape. Mathematical calculation gives the axis, and surface analysis gives the shape. These data leads to a part of the numerical notation [2].

Determination of Rotational Quart - Calculation of 2D Orientation. From the binary image and according to the bounding box, the system computes the principal axis of the binary shape. As described in [6] we use a mathematical method to determine the athlete axis. For discret $2D$ image probability distributions, the mean location (the centroid) within the search window, that is computed at step 3 above, is found as follows:

Find the zeroth moment :

$$A = \sum_x \sum_y I(x, y).$$

Find the first moment for x and y :

$$M_x = \sum_x \sum_y xI(x, y), \quad M_y = \sum_x \sum_y yI(x, y).$$

Mean search window location (the centroid) then is found as

$$\bar{x} = \frac{M_x}{A}, \quad \bar{y} = \frac{M_y}{A}.$$

The 2D orientation of the probability distribution is also easy to obtain by using the second moments in the binary image, where the point (x,y) ranges over the search window, and I(x,y) is the pixel (probability) value at the point (x,y).

Second moments are :

$$M_{xx} = \sum_x \sum_y x^2 I(x, y), \quad M_{yy} = \sum_x \sum_y y^2 I(x, y), \quad M_{xy} = \sum_x \sum_y xyI(x, y).$$

Let

$$a = \frac{M_{xx}}{A} - \bar{x}^2, \quad b = 2\left(\frac{M_{xy}}{A} - \bar{x}\bar{y}\right), \quad c = \frac{M_{yy}}{A} - \bar{y}^2.$$

Then the object orientation, or direction of the major axis, is

$$\theta = \frac{\arctan\left(\frac{b}{a-c}\right)}{2}.$$

The first two eigenvalues, that is, length and width, of the probability distribution of the blob found by the block filtering may be calculated in closed form as follows:

Then length l and width w from the distribution centroid are

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}, \quad w = \sqrt{\frac{(a+c) - \sqrt{b^2 + (a-c)^2}}{2}}.$$

When used in human tracking, the above equations give body roll, length, and width as marked in the source video image in Figure 3.

Rotation Tracking. The computed axis leads to the body orientation but it is not oriented. The given orientation is available at $\pm\pi$. Indeed, the function arctan is defined on $] - \frac{\pi}{2}; \frac{\pi}{2}[$. We use biomechanical constraints to eliminate ambiguous measurements. A study on physical constraints in acrobatics defined maximum angular velocities for twist and somersaults. The maximum velocity for somersault is $\omega_{max} = 22 \ rad.s^{-1}$ (or $\omega_{max} = 0.89 \ rad/image$ for videos running at 25 $images/s$). Variations around π are physically impossible. Thus the instantaneous angular velocity is : $\omega = \theta_{t-1} - \theta_t \ [\pi]$. And the correct orientation θ' is : $\theta'_t = \theta'_{t-1} + \omega$. This is a relative orientation depending on the first orientation.

Fig. 3. Human body's orientation, length and width

Quarters Detection. Find correctly the number of quarters of somersaults is not so obvious. When a gymnast execute a somersault, it is frequent that the body axis does not describe a entire rotation for a simple somersault (for a *f* 4 0 *o* the total rotation is $\theta'_t < 2\pi$). However, the system has to detect 2π i.e. 4 quarters of transversal rotation because the acrobat starts from feet and arrives on feet (figure 4). For example, a backward somersault (*b* 4 0 *o* in clockwise) starts at $\frac{\pi}{4}$ and finishes at $\frac{7\pi}{4}$. The number of quarters is 3 whereas the system has to detect 4. We introduce 3 methods to detect quarters. The first method is to calculate average angular velocity on the portions of somersault where this velocity is constant. A null angular acceleration generates a constant speed. While calculating the mean velocity of the constant phase one manages to have a good idea of the salto carried out. By deferring the mean velocity on the unit of the jump, we compensate the orientation problems of departure and the end of the jump. Another method is to compare the takeoff orientation with the landing orientation. This gives an exact measure. The last method consists in counting the number of dial crossed by the axis of the body. Somersault rotations are represented by a disc cut out in dials. This disc is divided into four zones ($[0; \frac{\pi}{2}[, [\frac{\pi}{2}; \pi[, [\pi; \frac{3\pi}{2}[, [\frac{3\pi}{2}; 2\pi[)$. To each time the axis enters a new dial it is entered. These three methods lead to the right detection. The differences among three methods of detecting quarters are : The first method gives sometimes too quarters due to the inertia of fast somersaults. However the second method is
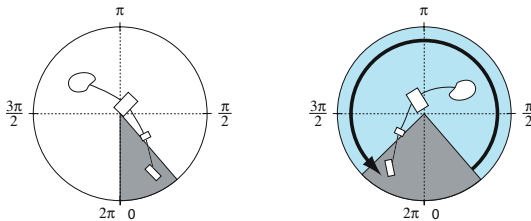
Fig. 4. Quarters rotation determination

concerned with the problem of the number of effective quarter and thus gives less quarters. The third method is correct around ± 1 quarter (when the acrobat starts from his back, he is hidden by the trampoline, this lead to miss quarter). By calculating the average of the three methods, quarters are correctly detected (table 1).

Table 1. Evaluation of recognition and tracking algorithm

Type	Test	Evaluation
Tracking correlation	93 %	94 %
Tracking standard deviation	26 pixels	26 pixels
Rotations	100 %	98 %
Positions o	57 %	50 %
Positions <	50 %	54 %
Positions /	100 %	100 %

Position Evaluation. The evaluation of the body shape, leads to *tuck*, *pike*, or *layout.* To this goal, the system needs the axis l, w and the surface of the gymnast (wich are variable size during somersault). The average l_m of the axis l and w is the diameter of a disc C centred on the centroid of the acrobat. The more the acrobat is tuck, the more the ratio of w on l is close to 1. And the more the ration of the surface of the disc C and the surface gymnast is close to 1. Most of the acrobat's pixels is included in the disc C. When these results are close to 0, we can conclude that the body shape is layout. This first stage makes it possible to differentiate tuck from layout. Nevertheless, tuck and pike are similar shapes. The pike shape is less compact than the tuck shape. Currently, the system is not able to make the difference between these two body shapes. Because acrobats are not perfect, these body shapes are not carried out perfectly. Compared to the rotational quarters' part, we have a similar problem and we have to discriminate right shapes.

Twist Evaluation. Twists are not detected yet. The recognition of all elements in the numerical notation is not complete. The twist detection is undoubtedly the most complex part to realize because this movement mix both transversal rotations and longitudinal rotations.

4   Experimental Result

To assess our system we carried out a video corpus which we manually labelled in part. We pinpointed the position of the head, hands, base, knees, and feet. The video corpus comprises more than 100 sequences and more than 1000 figures performed by 7 athletes. We divided the corpus into two parts: one for the algorithms adjustment and the other to evaluate them. Evaluations presented below were carried out on sequences with no audience in the background.

Body Tracking. From our labelled data, we deduced the centroid of the athlete (centroid of all labelled body parts). We calculated the correlation between the data measured manually and the data calculated automatically by the system. The result is presented in the table 1.

Rotationnal Quart Recognition. Quarters were counted manually and the counts were then compared with those calculated automatically. The percentage in table 1 represents the relationship between the quarters manually counted and those recognized by the system.

Position Recognition. In the same way, each body shape was evaluated. They were first manually labelled and the results were then compared with those recognized by the system.

Tracking results are sufficient to obtain a good recognition of the somersaults. The detection of quarters depends on the environment. The system limits appear clearly. Indeed it is not yet possible to use this system in a competition context, with a frantic audience in the background! In that situation, the recognition rate decreases shortly. A better primitives extraction algorithm is then required. The results for the recognition of quarters show that the system is effective, despite being occasionally mistaken in extreme cases. Recognition of the body shape has not been successful so far because it is not possible yet to discriminate between tuck and pike. The system regularly confuses the two shapes because they are relatively similar. However information which the system extracts already makes it possible to be exploited. Under training conditions the system shows a very good robustness. We will see in the section 5 that it was used in real conditions to adapt the sporting training.

As for processing time, the system runs in real time. After an initialization stage, one image is completed on average in 0.019 seconds on a 2.6 Ghz PC. The real time allows coaches to effectively use this system.

5   Training Evaluation

The system suggested in this article is not finalized. However it is able to extract significant information from training. In gymnastic apparatus such as the bars (high bar and uneven bars), the rings, the trampoline, the beam, the system brings many information which are not obvious during training. The system helps the coach on not easily remarkable information. We present an example during trampoline training.

We use the system to adapt the trampoline training. During competitions preparation, coaches prepare with gymnasts, sequences of 10 elements which will have to be perfect. An element is a jump, a skill (a salto for example). The ideal one is to carry out these 10 elements with a constant height. Figures 5 and 6 illustrate the amplitude variations of the 10 elements of a sequence. Each element is described by using the numerical notation quoted previously. The development of these sequences is not simple. Each gymnast has its characteristics and its facilities in realising elements. A movement can be appropriate to an athlete but
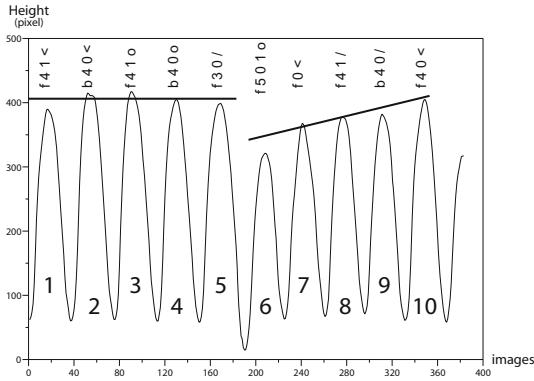
Fig. 5. Initial sequence (not adapted)
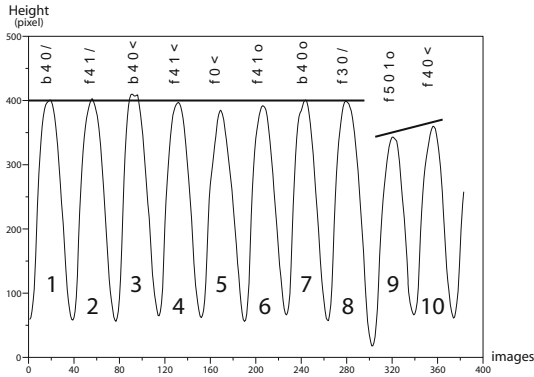
Fig. 6. Corrected sequence (adapted)

not for another. The example below illustrates a traditional sequence carried out by many gymnasts. During training, this sequence passes with difficulty for an individual. The gymnast has difficulties practising the movement. The traditional error would be he has to practice many more. However by looking at the figure 5 one notes a first relatively constant part (elements from 1 to 5), a setback (with the 6th element) and an increase towards the initial height until the end of the movement. The connection 5th element, 6th element is too difficult to realize at this place of the sequence. The idea is to move this connection at the end of the sequence. The system evaluates the sequence again.

In the corrected movement (figure 6) one finds this abrupt loss of amplitude (element 8 to 9). But this difficulty does not interfere any more with the other elements.

The athlete is an actor and can not see what is going wrong; it is the role of the coach. This one cannot be attentive with all details, especially when they are not easily locatable. The system highlights a considerable loss of amplitude which leads to a sequence mediocrity. The accused element then is replaced or moved.

The system evaluates the amplitudes variations on this example. It is also able to evaluate the velocities and acceleration of translation and angular rotations of the global human body. This information is not easily reachable by the human eye. In the next improvements, the system should be able to recognize and evaluate them.

6   Perspectives and Conclusion

The first prospect for the system is to be able to evaluate twists quantitatively. The use of optical flow could solve this problem. The first measurements of the optical flow [7] leads us to pursue our investigations. The optical flow remains an extremely expenditure in computing time and could offset the value of real time computing. We therefore propose to calculate the optical flow on parts of the bounding box after having restored the axis of the body to a vertical position. The rotational component would be cancelled out following the transversal axis and the translational component. This calculation should highlight only the longitudinal rotational component. The second prospective element is to finalize the system to make it a robust recognition system for acrobatic gesture.

This paper presents an analysis system of sporting gestures by global measurements. The system does not identify the parts of the human body but bases its recognition on measures of the global human body. Such a system is taken out of laboratories because it makes analysis in sport context and not in laboratories context. The system is not intrusive. There are no constraints for the sportsman (no sensor obstructing), it preserves the naturality of the analyzed gesture. The simplicity of implementation and the low cost of the hardware make the system accessible to sports coaches. Global measurement can lead to a robust recognition thanks to an adequate characterization. The low complexity of the algorithms allows real time. The system gives useable results and it is already used for training. The system reaches its limits when the camera is not fixed or when a crowd of people is moving in the background. In the same way the system does not allow analysis of two persons in the field of the camera. The system recognizes acrobatic gesture by global measurement. The terminology employed is effective and is recognized in the international trampoline community. It is certainly not very pleasant but each element has a translation in every language. This kind of system is very helpful for coaches and judges in competition.

References

1. H. Lakany and G. Hayes, "An algorithm for recognising walkers," *Second IAPR Workshop on Graphics Recognition*, Nancy, France , pp. 112-118 , August 22-23, 1997.
2. Ryan Cassel and Christophe Collet, "Tracking of Real Time Acrobatic Movements by Image Processing," *5th International Gesture Workshop*, Genova, Italy, pp. 164 - 171 , April 15-17, 2003.

3. A. Braffort, A. Choisier, C. Collet, et al., "Toward an annotation software for video of Sign Language, including image processing tools and signing space modelling," *4th International Conference on Language Resources and Evaluation*, Lisbonne, Portugal, 2004.

4. Masanobu Yamamoto, Takuya Kondo, Takashi Yamagiwa and Kouji Yamanaka, "Skill Recognition," *3rd IEEE International Conference on Automatic Face and Gesture Recognition,* pp.604-609, 1998.

5. Greg Welch, Gary Bishop, "An Introduction to the Kalman Filter," *TR 95-041*, Technical Report, Department of Computer Science, University of North Carolina, NC, USA, 2002

6. Ryan Cassel and Christophe Collet, "Caracterization and Tracking of Acrobatic Movements for Recognition," *Workshop in 14ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, Toulouse, France, 2004.

7. Changming Sun, "Fast Optical Flow Using 3D Shortest Path Techniques," *Image and Vision Computing*, Vol. 20, no.13/14, pp.981-991, December, 2002.

8. Gopal Pingali, Agata Opalach, Yves Jean, "Ball Tracking and Virtual Replays for Innovative Tennis Broadcasts," *International Conference on Pattern Recognition (ICPR'00)*, p. 4152, Volume 4, September 03 - 08, Barcelona, Spain, 2000.

9. Fanch Lejeune, Annelies Braffort and Jean-Pierre Descls, "Study on Semantic Representations of French Sign Language Sentences," *4th International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, P. 197-201, London, UK, 2001.

10. Kong Man Cheung and Simon Baker and Takeo Kanade, "Shape-From-Silhouette of Articulated Objects and its Use for Human Body Kinematics Estimation and Motion Capture," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June, 2003.

11. Fédération Internationale de Gymnastique, "Trampoline Code of Points 2005," www.fig-gymnastics.com.

Gesture Spotting in Continuous Whole Body Action Sequences Using Discrete Hidden Markov Models

A-Youn Park and Seong-Whan Lee

Department of Computer Science and Engineering, Korea University,
Anam-dong Seongbuk-ku, Seoul 136-713, Korea
{aypark, swlee}@image.korea.ac.kr
http://image.korea.ac.kr/

Abstract. Gestures are expressive and meaningful body motions used in daily life as a means of communication so many researchers have aimed to provide natural ways for human-computer interaction through automatic gesture recognition. However, most of researches on recognition of actions focused mainly on sign gesture. It is difficult to directly extend to recognize whole body gesture. Moreover, previous approaches used manually segmented image sequences. This paper focuses on recognition and segmentation of whole body gestures, such as walking, running, and sitting. We introduce the gesture spotting algorithm that calculates the likelihood threshold of an input pattern and provides a confirmation mechanism for the provisionally matched gesture pattern. In the proposed gesture spotting algorithm, the likelihood of non-gesture Hidden Markov Models(HMM) can be used as an adaptive threshold for selecting proper gestures. The proposed method has been tested with a 3D motion capture data, which are generated with gesture eigen vector and Gaussian random variables for adequate variation. It achieves an average recognition rate of 98.3% with six consecutive gestures which contains non-gestures.

1 Introduction

Human gesture recognition has a wide range of application such as human-machine interaction, surveillance, machine control etc[1,2]. For these applications, it is necessary to develop efficient and automatic gesture recognition and segmentation algorithm. In early works, many researchers has bee studied as an alternative form of human-computer interface by Starner and Pentland[3] and Quek[4]. Pentland used a Hidden Markov Model(HMM) to recognize the gestures in American Sign Language. They achieved an accuracy of 99.5 % for 15 gestures. However, this approach is not suitable to recognize whole body daily gestures because the syntax is not well understood. Moreover, they used manually segmented image sequences. Kahol[5] proposed whole body gesture segmentation algorithm in dance sequences. They employed a dynamic hierarchical layered structure to represent human anatomy. This method used a few data to test the algorithm and they focused on only segmenting gestures. Like these works, most approaches of gesture recognition using HMM used manually segmented image sequences so that it is difficult to extend to apply to continuous gesture recognition.

In this paper, we focus on recognition of whole body gesture and spotting a meaningful gesture. We describe gesture as a spatio-temporal sequences of multi-dimensional features from gesture sequences and cluster the features using Gaussian Mixture Model(GMM) for HMM input. The frame sequences are recognized as gestures using the probability calculation of HMM and segmented meaningful gestures using gesture spotting algorithm. For rejecting of non-gestures, each reference pattern is defined by a keyword model and all the other patters are modeled by garbage model which can be used as a threshold value. Figure 1 shows the block diagram of the proposed gesture spotting method.
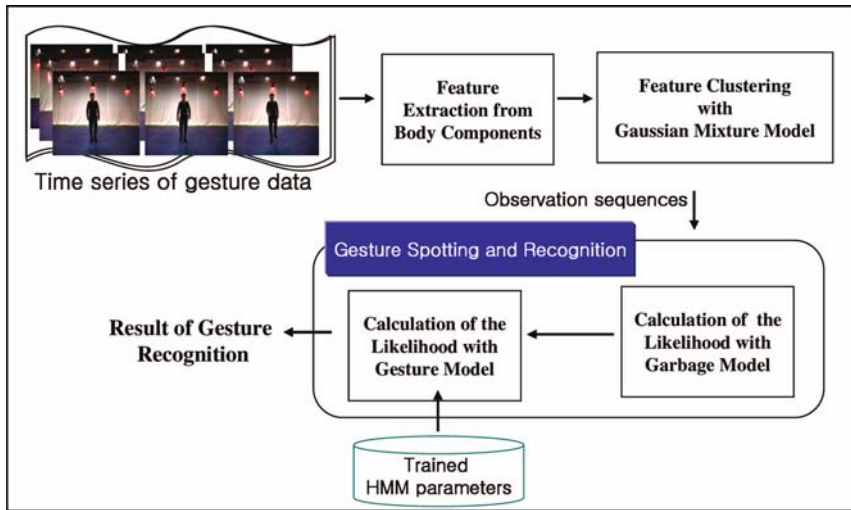
Fig. 1. Block diagram of the proposed gesture spotting method

2   Gestures Modeling

We will describe which feature is good for our purpose in following section. After extraction the features, we represent each gesture as a sequence of body features. The temporal relation between these feature is enforced by a hidden markov model(HMM), which will be presented in section 3.

2.1   Feature Extraction from Body Components

Before extracting feature, we detect and track each body component in each image sequence at the pre-processing stage using Yang[6]. He proposed a 3D human body pose reconstructing method(see Fig 2). We assume each body component is detected and tracked using 3D modeling with few errors at the pre-processing.

Once each body component is detected, we extract the features from them. On of an important cue in discriminating each gesture is to choose a good feature. Raw position (x, y, z), and the Catesian velocity (dx, dy, dz) can be used as a features. However, raw position is sensitive to rotation and translation. Cartesian velocity is

Fig. 2. Each component detecting in image sequence using 3D modeling

invariant to translation but is also sensitive to rotation. We choose the angle between the mid-point and each component for invariance of rotation and translation (see Fig. 3). We use the following thirteen components: mid-point, left-right shoulder, elbow, wrist, hip, knee, and, ankle. Since there are thirteen components, the feature vector can be represented as *F* in each image sequences.

$$F = \{\theta_{mid\_point}, \theta_{left\_shoulder}, \theta_{right\_shoulder}, \theta_{left\_Elbow}, \theta_{right\_Elbow}, ..., \theta_{right\_ankle}\}$$

Each frame is expressed by the feature in high dimension at that particular time. In this manner, gesture is defined as an ordered sequence of feature vector, *F*. We projected this feature vector high dimension and cluster these features. Each cluster will be described with $< \mu_i, \Sigma_i >$ where center matrix, and covariance matrix in ith cluster respectively.
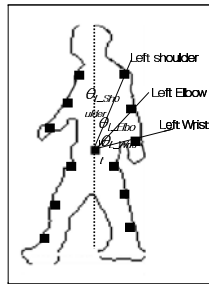
Fig. 3. Features from each body component

2.2 Feature Clustering

To state the problem mathematically, given a long motion sequence, *M* , we wish to segment that sequence into distinct gesture $M_1, ..., M_S$ where *S* is the number of boundaries of the gestures(see Fig 4). Each gesture sequence *M* is represented as a sequence of body features, *F*, which are angles between mid back and each component

$$M = \{M_1, M_2, ..., M_s\}$$
$$M_s = \{X_1, X_2, ..., X_n\}$$
$$X_n = \{\theta_1, \theta_2, \theta_3 ... \theta_f\}$$

where *s* is the number of boundaries of gesture, *n* is the number of frames, and *f* is the number of features in a single image.

Fig. 4. Several continuous gestures sequences

In the first phase, we learn the distribution of the data distributions without temporal information. We cluster each feature vector to model gesture as cluster trajectories.

The underlying assumption of cluster approach is that the frames from different simple motions form separate clusters and each cluster can be described reasonably well by a Gaussian distribution. Moreover, the different gesture has different cluster trajectories even though different simple motion is in the same cluster. We employ a Gaussian Mixture Model(GMM) to cluster the features so that each motion sequence corresponds to the trajectory in cluster. It means that gesture can be modeled by consecutive cluster trajectories belong to different elements of a GMM, which groups set of frames in different Gaussian distribution.

We use the Expectation Maximization (EM) algorithm[7] to estimate the Gaussian Mixture Model of the data. Each cluster,C, indicates a region in the high dimension space that is represented by the centroid, μ_s, and covariance matrix ,Σ_s. Given input data vector F, the distance from the data to the state S is define Mahalanobis distance.

In the ideal case, each cluster is represented by a single Gaussian distribution. Thus, a collection of k clusters can be represented by a mixture of k Gaussian distribution. Note that the clusters are not of equal size. After, the GMM parameters are estimated, we compute a most likely cluster for each frame of the motion and the index of most likely cluster is observation symbol in HMM.

Each gesture has its own gesture cluster index sequences. We show the examples of gesture cluster index sequences. The different gesture has different cluster trajectories, on the other hand, same gesture has very similar trajectories

$$Walking\_Gesutre = \{2,4,4,4,4,1,3,1,...,3,4,6,6,....9,9,...4,4,4,.1..9\}$$
$$Running\_Gesture = \{24,8,7,2,2,1,6,...3,3,3,3,6,..19,18,5,...17,17,19\}$$

We have to specify the number of clusters, *C*, for each execution of the GMM, and we usually do not know the number of clusters in a data set. We fit it with *C=25*. This number seem to be useful for our data and fixing *C* such as 25 seemed to produce reasonable results without additional complexity.

3   Gesture Recognition and Segmentation

The main requirement for the use of gesture recognition in human-computer interfaces is a continuous online recognition with a minimum time delay of the recognition output (see Fig. 5).
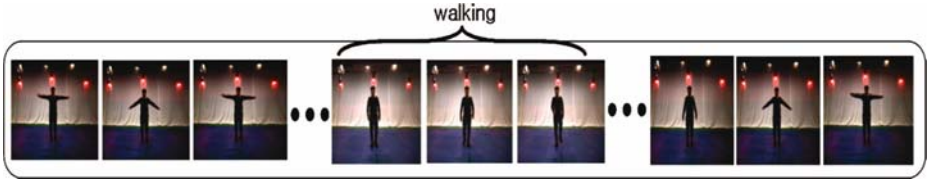
Fig. 5. Spotting gestures in a continuous gesture stream

However, such a system performance is extremely difficult to achieve, because the fact that the starting and ending point of the gestures is not known and it is a very complicated problem. This paper focus on the automatic finding the starting and ending point, called gesture spotting. Our method works with continuous gesture streams and allows an automatic temporal segmentation and recognition at a the same time.

3.1   Gesture Recognition

The HMM is rich in mathematical structures so that it has been widely used for many classification and modeling problems, especially, in the speech recognition and handwriting recognition. We choose the HMM-based approach because it can be applied to analyze time-series with spatio-temporal variations and can handle undefined patterns effectively.

The HMM is a collection of states connected by transitions. Each transition has a pair of probabilities: a *transition probability* and an *output probability.* Following Rabiner paper [8], a compact notation $\lambda = \{$ A, B, $\Pi\}$ is used which includes only probabilistic parameters. Every Gesture is represented by each HMM probabilistic parameters, $\lambda_{1,\cdots M}$ where *M* is the number of gestures

The most general approach uses a fully-connected model. However, training of these models leads to ambiguous model with high entropy, not suitable for production purpose so that we design a model using the left-right HMM utilizing the temporal characteristics of gesture signals for each gesture. The left-right model is good for modeling order-constrained time-series whose properties sequentially change over time. The number of states in a model is determined based on the complexity of the corresponding gesture. The number of states in our gesture models ranges from five to eight, depending on the complexity of the gesture. The gesture models are trained using Baum-Welch reestimation algorithm.

To recognize observed symbol sequences, we create HMM for each gesture. We choose the model which best matches the observations from gesture HMM λ_c . This means that when a sequence of unknown category is given, we calculate the $P[(M \mid \lambda_c]$ for each gesture and chose the HMM that has the highest value. Each gesture is recognized in following equation.

$$Gesture = \arg \max_{c} [\, p(M \mid \lambda_c)\,] \tag{1}$$

where c is the number of the gestures and M are observation sequences.

3.2 Key Gesture Spotting Network

We construct the HMM gesture spotting network with two garbage models and one gesture HMM model. The first garbage model is in front of a gesture HMM to reject the non sub-pattern gestures which start before meaningful gesture. One gesture HMM and second garbage model is constructed together.

For correcting gesture spotting, the likelihood of a gesture model, which is mentioned previous section, for a given pattern should be distinct enough. Unfortunately, although the HMM recognizer chooses a model with the best likelihood, we cannot guarantee that the pattern is really similar to the reference gesture unless the likelihood value is high enough. Therefore, we propose a garbage model that gives confidence measure to reject the non-gesture.

The garbage models are made by gesture states. The HMM's internal segmentation property implies that each state with its self-transition represents a segmental pattern of a target gesture and that outgoing transitions represents a sequential progression of the segments in a gesture.

With this property, we can construct an ergodic model with the states copied from all gesture models in the system and then fully connect the state (see Fig 6). We construct our garbage model as following step.

First step: Self-transition probabilities are kept in the gesture models.

Second step: Output observation probabilities(b_j(k)) are copied from gesture models and we reestimate that probabilities with gaussian distribution smoothing. Gaussian smoothing of the output probabilities distribution makes the states represent any patterns.

$$G(\mathbf{b}_j(k)) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(b_j(k))^2}{2\sigma^2}} \qquad (2)$$

Third step: all outgoing transition probabilities are equally assigned as

That equation means each state is reached by all other possible state in a single transition so that this garbage model is erogodic model which makes it match well with any patterns generate by combining the sub-patterns in any order.

$$a_{new-ij} = \frac{1 - a_{old-ij}}{N-1}, \quad \text{for all } j, i \neq j. \qquad (3)$$

where a_{new_ij} is the transition probabilities of garbage model from state s_i to state s_j, a_{old_ij} is the transition probabilities of gesture model from state s_i to state s_j, and N is the number of all gesture state. The start and final state produce no observation.

The garbage model is used as a confidence measures for rejecting a non-gesture pattern. The confidence measure can be calculate using the garbage-model as an approximate of P(X)

$$P(X) = P(X \mid \lambda_{grabage\_model}) \qquad (4)$$
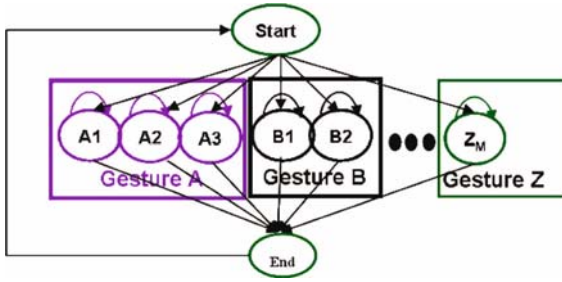
Fig. 6. The garbage model
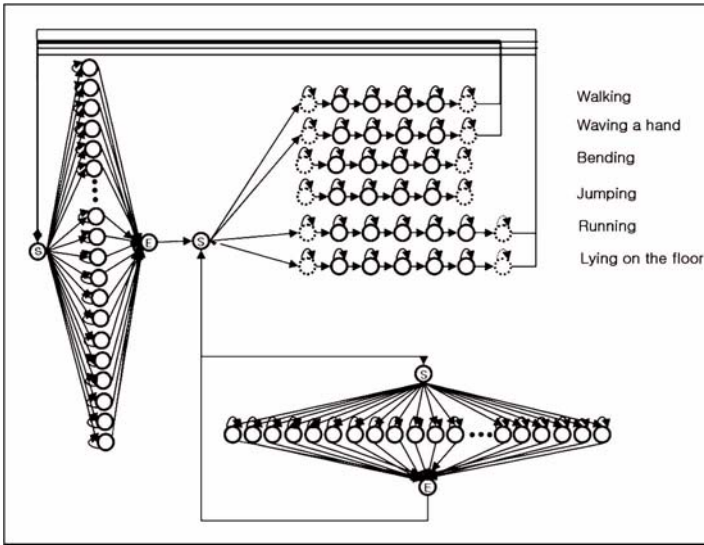
Fig. 7. Gesture spotting network

where *X* are an input observation sequences, and $\lambda_{garbage\_model}$ is an HMM parameters of garbage model. Our HMM network is shown in fig. 7. The method, which spots a meaningful gesture pattern, will be described in next section.

To find the single best state sequence, $Q_{1,t} = q_1, q_2, ..., q_t$ for the given observation $O_{1,t} = O_1, O_2, ... O_t$ we need to define the quantity:

$$\delta_t(i) \cong \max_{Q_{1,t}} P(Q_{1,t-1}, q_t = s_i, O_{1,t} \mid \lambda)$$

with the highest probability along a single path arriving at s_i at time t and accounting for the first observation. If the sequence is non gesture pattern, it is filtered by the first garbage model.

For the backtracking information, we use $\varphi_t(j)$ to keep the argument that maximizes it for each t and j

$$\psi_t(j) = \arg\max_i [\delta_{t-1}(i)a_{ij}] \qquad 2 \le t \le T, \ 1 \le j \le S. \tag{6}$$

where S is the number of state, and T is total time of input gesture.

In case of null transitions,, the likelihood of the source state at time t is simply maximized without time delay as

$$
\begin{aligned}
\delta_t(j) &= \max_i [\delta_t(i)a_{ij}], \\
i^* &= \arg\max_i [\delta_t(i)a_{ij}], \\
\psi_t(j) &= \psi_t(i^*).
\end{aligned}
\tag{7}
$$

Finally, to uncover the most likely state sequence Q*=q*1q*2…q*T after the preceding computation, we must trace back to the initial state by following the Viterbi path. This algorithm described is known as *Viterbi algorithm*, and we refer the reader to [9] for algorithm details.

$$
\begin{aligned}
q^*_T &= s_N, \\
q^*_t &= \psi_{t+1}(q^*_{t+1}) \qquad t = T-1, T-2, \dots 1.
\end{aligned}
\tag{8}
$$

with the highest probability along a single path arriving at s_i at time t and accounting for the first observation. If gesture pattern starts, the log likelihood value of the gesture HMM is above of confidence measure($P(X|G_{garbage\_model})$). We can use confidence measures as threshold.

4   Experiments and Analysis

4.1   Experimental Data

The six gestures form the basis of the test gesture. These gesture data are selected from FBG database[10]. The FBG database contains 14 representative full-body gestures in daily life for 20 performers. This database consists of major three parts: 3D motion data, 2D stereo-video data and 2D silhouette data. The database has abundant variation.

However, these data do not have enough variation to test our algorithm so that we generate the gestures. We calculated the eigen gesture vector per each gesture with 20 performers using principal component analysis (PCA). After that, we generate the Gaussian random coefficient value and the combine the eigen gesture vector and Gaussian random coefficient value linearly (see fig. 8).

4.2   Results

We divide the gesture data into 50 training and 50 test data. The six gesture HMMs are trained with the isolated training gestures.

In this method, we test the continuous gesture. The detection ratio is the ratio of correctly recognized gestures over the number of input gestures for evaluation of recognition algorithm:

$$\text{Detection ratio} = \frac{\text{\# of correctly recognized gestures}}{\text{\# of input gestures}} \tag{8}$$
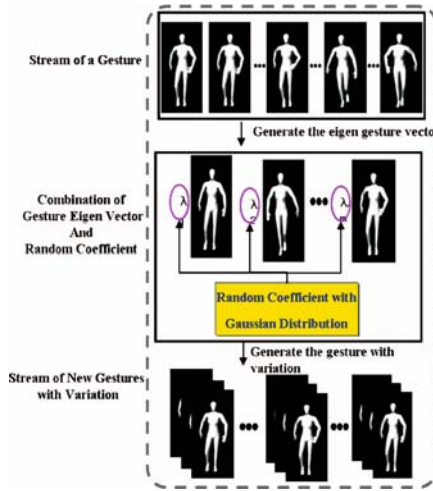
Fig. 8. A diagram of generating the various gesture with gesture eigen vector

We fail to recognize two running gesture because the gesture is smoothed too much so that the gesture is very similar to walking gesture. Our proposed method achieves an average recognition rate of 98.3%.

The table 1 gives the average recognition accuracy obtained continuous gesture.

Table 1. The result of gesture recognition

Gestures	The number of input gesture	The number of Correctly recognized gesture	Detection ratio
Walking	50	49	98%
Running	50	48	96%
Bending	50	50	100%
Jumping	50	49	98%
Lying down on the floor	50	50	100%
Waving a hand	50	50	100%

In addition to gesture recognition, we also test the spotting algorithm in the continuous gesture input streams (see fig. 9).

A time-evolution of the likelihood of individual model is shown in Fig. 10. From 0 sec. to 9 sec., the likelihood of garbage model is high than that of other gesture models. There are no any meaningful gestures in this period so we reject this gesture. After time 9 sec., however, the likelihood of walking gesture model becomes the greater than that of the garbage model. When the likelihood of the key gesture model

(a) Non gesture sequences from 0 sec. to 9 sec.

(b) Walking gesture sequences from 9 sec. 19 sec.

(c) Bending gestures sequences from 19 sec. to 25 sec.
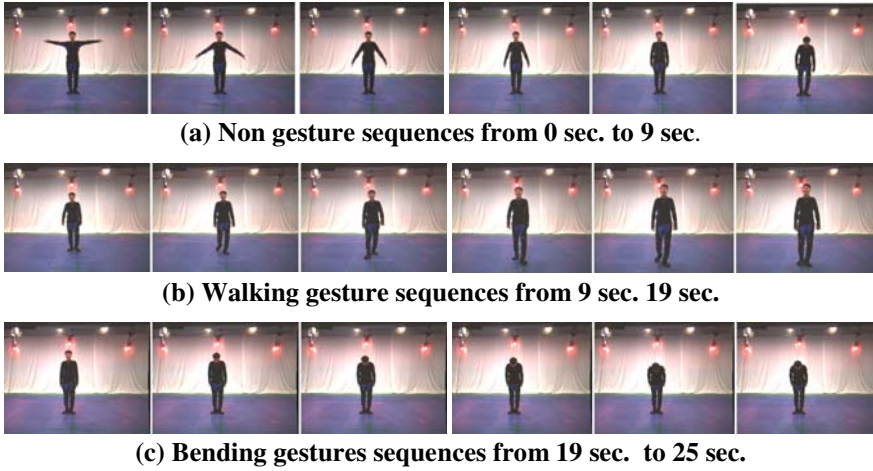
Fig. 9. Continuous image streams with non-gesture and meaningful gestures
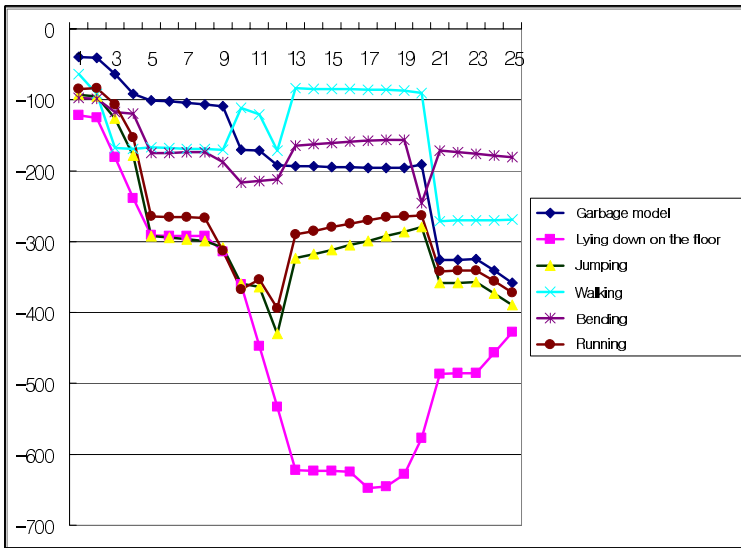
Fig. 10. The likelihood evolution of the gesture models and the garbage model

is above that of garbage model, the starting point is determined by backtracking the *Viterbi* path. This key gesture is segmented in this time period. Finally, the likelihood of bending gesture model from 19 to 25 is above other gesture models. In this period, we also segment bending gesture. We segment two different gesture, walking and bending gestures, and reject non-gestures. We measure our spotting algorithm evaluation with reliability. The reliability is introduced that considers the insertion errors as follows.

$$\text{Reliability} = \frac{\text{\# of correctly recognized gestures}}{\text{\# of input gestures} + \text{\# of insertion errors}} \quad (10)$$

The insertion error occurs when the spotter reports a nonexistent gesture. The deletion error occurs when the spotter fails to detect a gesture. The substitution error occurs when the spotter falsely classifies a gesture. Table 2 shows the spotting performance with reliability.

Table 2. Spotting results with reliability

Gestures	The number of input gesture	The number of Correctly recognized gesture	Delete errors	Substitute errors	Insertion errors	Reliability
Walking	58	55	0	1	2	91.6%
Running	62	57	1	1	3	91.9%
bending	54	54	0	0	0	100%
Jumping	62	61	0	0	1	96.8%
Lying on the floor	61	58	0	1	2	92.1%
Waving a hand	60	58	0	1	1	95.1%
total	357	343	1	4	9	93.7%

5   Conclusion and Further Work

This paper describes an HMM-based gesture recognition and segmentation with a GMM clustering and garbage model. The proposed method not only provides invariance with respect to the speed of the movement because HMM can deal with time-sequential data, but also covers the variation generated by multiple people because we use the cluster trajectory in which each cluster is described well by a Gaussian distribution. The garbage model also provides a good confirmation mechanism for rejecting the non-gestures.

In the future, we will extend to this algorithm to real image sequences. We will test more sophisticated and various body gestures. In addition, we embed this methodology into a robot so that the robot can recognize the human body gesture, react it and help the old people when they fall down unexpectedly.

Acknowledgement

References

1. Wilson, A.D., Bobic, A.F.,: Parametric Hidden Markov Models for Gesture Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 21, No. 9, (1999), 884-900

2. Vaananen, K., Boehm, K.,: Gesture Driven Interaction as a Human Factor in Virtual Environments – An Approach with Neural Networks., Virtual Reality Systems, (1993), 93-106

3. Starner, T., Weaver, J., Pentland, A.,: Real-Time American Sign Language Recognition Using Desk and Wearable Computer Based Video., IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 20, No. 12, (1998), 1371-1375

4. Quek, F.,: Toward a Vision-Based Hand Gesture Interface., Proc. of Virtual Reality System Technology Conf., Singapore, (1994) 17-29

5. Kahol, K., Tripath, P., Panchanthan. S.,: Automated Gesture Recognition From Dance Sequences, Proc. of Int'l Conf. on Automatic Face and Gesture Recognition, Seoul, (2004), 883-888.

6. Yang H.-D., Park S.-K., Lee S.-W.,: Reconstruction of 3D Human Body Pose Based on Top-down Learning

7. Duda, R., O. Hart, P.E. Stork, D. G.,: Pattern Classification. Wiley & Sons, New York (2001), Proc. of Int'l Conf. on Intelligent Computing, Hefei, (2005), To appear.

8. Rabiner., L. R.,: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition., Proc. of IEEE, Vol. 77 (1989) 257-286

9. Viterbi, A. J.,: Error Bounds for Convolution Codes and an Asymptotically Optimum Decoding Algorithm., IEEE Trans. on Information Theory, Vol. 13 (1967) 260-269

10. Hwang B.-W., Kim S.-M., and Lee S.-W.,: 2D and 3D Full-Body Gesture Database for Analyzing Daily Human Gestures., Proc. of Int'l Conf. on Intelligent Computing, Hefei, (2005), To appear.

Recognition of Deictic Gestures
for Wearable Computing

Thomas B. Moeslund and Lau Nørgaard

Laboratory of Computer Vision and Media Technology,
Aalborg University, Denmark
tbm@cvmt.dk

Abstract. In modern society there is an increasing demand to access, record and manipulate large amounts of information. This has inspired a new approach to thinking about and designing personal computers, where the ultimate goal is to produce a truly wearable computer. In this work we present a non-invasive hand-gesture recognition system aimed at deictic gestures. Our system is based on the powerful Sequential Monte Carlo framework which is enhanced with respect to increased robustness. This is achieved by using ratios in the likelihood function together with two image cues: edges and skin color. The system proves to be fast, robust towards noise, and quick to lock on to the object (hand). All of which is achieved without the use of special lighting or special markers on the hands, hence our system is a non-invasive solution.

1   Introduction

In modern society there is an increasing demand to access, record and manipulate large amounts of information involved in many aspects of professional and private daily life. This has inspired a new approach to thinking about and designing personal computers, where the ultimate goal is to produce a truly wearable computer. Wearable in the sense of being a natural extension of the body like clothes, shoes or glasses. A brief historic overview of wearable computing is listed below, see [11] for further details.

1268 Earliest recorded mention of eyeglasses
1665 Robert Hooke calls for augmented senses
1762 John Harrison invents the pocket watch
1907 Aviator Alberto Santos-Dumont commissions the creation of the first wristwatch
1960 Heilig patents a head-mounted stereophonic TV display
1960 Manfred Clynes coins the word "Cyborg"
1961 Edward O. Thorp and Claude Shannon (MIT) builds the first wearable computer, which is used to predict roulette wheels
1966 Ivan Sutherland creates the first computer-based head-mounted display (HMD)
1977 Hewlett-Packard releases the HP 01 algebraic calculator watch
1979 Sony introduces the Walkman
1981 Steve Mann designs backpack-mounted computer to control photographic equipment

1985 (Wearable) Device for prediction or card counting in casino games were outlawed in the state of Nevada

1989 Private Eye's HMD sold by Reflection Technology

1991 Doug Platt debuts his 286-based "Hip-PC"

1991 Carnegie Mellon University (CMU) team develops VuMan 1 for viewing and browsing blueprint data

1993 Thad Starner starts constantly wearing his computer, based on Doug Platt's design

1993 Feiner, MacIntyre, and Seligmann develop the KARMA augmented reality system

1994 Lamming and Flynn develop "Forget-Me-Not" system, a continuous personal recording system

1996 Boeing hosts "Wearables Conference" in Seattle

1997 CMU, MIT, and Georgia Tech co-host the first IEEE International Symposium on Wearable Computers

1997 First Wearable computer fashion show at MIT

1.1   Related Work

A number of different devices have been developed or adopted to the special user interface requirements in wearable computing [9]. Depending on the context the requirements differ. However, one common issue in most wearable computing interfaces is the need for a pointing device, similar to the computer mouse used in standard WIMP interfaces. Without it, precise deictic interaction is either not possible or very cumbersome.

The most common way of achieving this is by the use of a data glove, see e.g., [10]. Glove-based methods are by nature intrusive and besides often too expensive and bulky for widespread use [13]. Other intrusive devices which have been used to provide deictic input are bend sensors [13], ultrasonic devices [4], and accelerometers [13]. Less intrusive methods are based on head-mounted cameras segmenting the hand(s) in the image. Compared to some of the more intrusive devices cameras in general produce poor signal-to-noise ratios and therefore either infrared light and cameras, see e.g., [14] and [12], or markers on the hands/fingers, see e.g., [10], are used. For further information on state-of-the-art see [8].

1.2   The Content of This Paper

The aim of this paper is to develop a head mounted camera-based gesture interface for wearable computing that neither requires special lighting (infrared) nor markers attached to the hands/fingers. Our approach is to adopt an advanced tracking framework: the Sequential Monte Carlo (SMC) method [3], which is often used in the computer vision research field, see e.g., [5][1][2], and tailor it to the needs originating when the camera is head mounted.

The paper is structured as follows. In section 2 the gestures are defined and a representation is derived. In section 3 the tracking framework for the gesture recognition is presented. In section 4 and section 5 the recognition of the pointing gesture is described. In section 6 the system is tested and in section 7 a conclusion is given.

2   Modeling the Pointing Gesture

We require two gestures to be recognizable, pointing and clicking. The former is defined as an outstretched index finger and thumb with all other fingers bend. The latter is defined in the same way except that the thumb is now held against the index finger, see figure 1. As only the movement of the thumb is modeled explicitly and the other fingers are kept still or hidden, the range of internal motion is limited to the three joints of the thumb, see figure 1. As the DIP (Distal Interphalangeal) joint is always outstretched when doing the click gesture, and since the MCP (Meta CarpoPhalangeal) joint is difficult to bend independently of the CMC (Carpal MetaCarpal) joint, the two gestures can be represented and distinguished by one only one rotational DoF (degree of freedom).
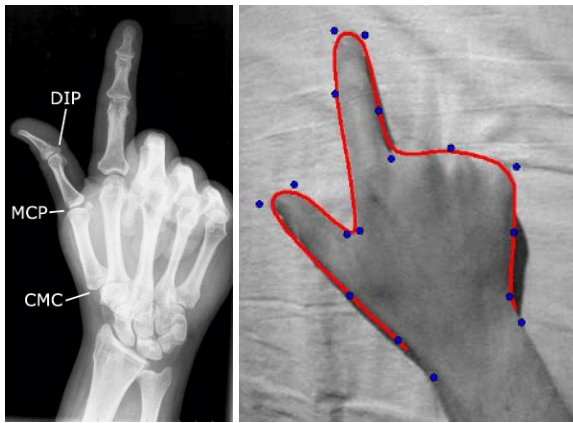
Fig. 1. A hand showing the point gesture. Left: X-ray image. Right: B-spline approximation of the contour. Control points are marked with circles.

We represent the appearance of the two gestures by the contour of the hand, see figure 1. The contour is manually created using B-splines defined by a set of control points. Applying a linear transformation to these points is the same as applying the transformation to the B-splines.

To create the set of two basis splines required to model the two gestures, a spline was first fitted to an image of the pointing gesture and saved to an ASCII file. Then an image with the hand in the same position but showing a clicking gesture was loaded, and the control points moved to make the spline follow the thumb. These two splines were then loaded into the tracker and combined using linear interpolation. So one parameter, ϕ, controls the internal DoF in our hand model.

Regarding the external DoF for the hand model, we assumed weak perspective transformation and applied an affine transformation. So the final external transformation of the B-spline contour is given as:

$$\mathbf{r}(s) = \begin{bmatrix} d_x \\ d_y \end{bmatrix} + \begin{bmatrix} s_x\,cos(\theta) & a\,s_y\,cos(\theta) - s_y\,sin(\theta) \\ s_x\,sin(\theta) & a\,s_y\,sin(\theta) + s_y\,cos(\theta) \end{bmatrix} \mathbf{r}_0(s) \qquad (1)$$

where $\mathbf{r}_0(s)$ is the set of untransformed control points representing the contour of the hand, $\mathbf{r}(s)$ is the set of affine transformed control points, d_x and d_y define the translation in the image plane, s_x and s_y define the scaling in the image plane, a is the shear, and θ is the rotation in the image plane.

In total we ended up with 7 DoF, one internal and six externals. That is, our state-space is seven-dimensional and one configuration of the hand is defined by the state vector, $\mathbf{x}$, as:

$$\mathbf{x} = (d_x, d_y, \theta, s_x, s_y, a, \phi) \tag{2}$$

where ϕ is the angle between the index finger and the thumb.

3   Tracking Framework

Due to the low signal-to-noise ratio mentioned in section 1.1 the hand can not always be segmented perfectly from the background. Hence, the recognition in a particular frame will not always be unique, or in statically terms, the conditional probability of a gesture given the image measurements will in general be multi modal. This calls for a Sequential Monte Carlo (SMC) method which can handle such situations [3]. The SMC algorithm operates as most other tracking frameworks, by using a predict-match-update structure.

The SMC is defined in terms of Bayes' rule and by using the first order Markov assumption. That is, the posterior PDF (probability density function) is proportional to the observation PDF multiplied by the prior PDF, where the prior PDF is the predicted posterior PDF from time $t - 1$:

$$p(\mathbf{x}_t|\mathbf{u}_t) \propto p(\mathbf{u}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{u}_{t-1}) \tag{3}$$

where $\mathbf{x}$ is the state and $\mathbf{u}$ contains the image measurements. The predicted posterior PDF is defined as

$$p(\mathbf{x}_t|\mathbf{u}_{t-1}) = \int p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{u}_{t-1}) \, d\mathbf{x}_{t-1} \tag{4}$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ is the motion model governing the dynamics of the tracked object, i.e., the prediction, and $p(\mathbf{x}_{t-1}|\mathbf{u}_{t-1})$ is the posterior PDF from the previous frame.

This formula is exactly what we are after, as it imposes no constraints on the posterior PDF, as for example the Kalman filter does. However, even with a coarse resolution for the different parameters in the state vector (both internal and external DoF), too many combinations exist and it is not computationally feasible to evaluate the integral in equation 4. Therefore, the SMC algorithm approximates the posterior by only sampling the N most appropriate combinations. In praxis this is done by estimating $p(\mathbf{x}_t|\mathbf{u}_t)$ by selecting a number, N, of (hopefully) representative states (particles) from $p(\mathbf{x}_{t-1}|\mathbf{u}_{t-1})$, predicting these using $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, and finally giving each particle a weight in accordance with the observation PDF, $p(\mathbf{u}_t|\mathbf{x}_t)$. For the next time step N new particles are drawn from the existing set with probabilities proportional to the calculated weights.

This approach will concentrate most particles around the likely hypotheses, but since the prediction contains both a deterministic and a stochastic element, some particles will also spread out randomly making the SMC algorithm able to cope with unpredicted events.

4   Motion Model

In order to apply the SMC method we need to be able to predict a state vector over time. In this section the motion model, which implements the prediction, is defined.

We assume the individual dimensions in the state space are independent, and can be modeled by first order auto-regressive (AR) processes:

$$x_t - \overline{x} = a(x_{t-1} - \overline{x}) + bw_t \Leftrightarrow x_t = \overline{x} + a(x_{t-1} - \overline{x}) + bw_t \qquad (5)$$

where x_t is the value at time t, $\overline{x}$ is the mean or expected value and w_t is a random variable with distribution $\mathcal{N}(0, 1)$.

The only exception is x and y translation, d_x and d_y, which are modeled by second order AR processes with constant velocity:

$$x_t = x_{t-1} + (x_{t-1} - x_{t-2}) + bw_t \qquad (6)$$

Consequently there are 12 motion constants to be determined during training. In addition, the mean and standard deviation for each dimension are also calculated in order to determine the *a priori* distributions used in the initialization of the SMC tracker [9].

The motion model and initialization assume the value and step size along each dimension of the state space to be normally distributed except d_x and d_y which are assumed uniform. Test justify these assumptions except for the value of the thumb angle, ϕ, which is not normally distributed [9]. Its histogram has two distinct modes at positions corresponding to the point and click gestures. We handle this by modeling the two modes by two first order AR processes, as in equation 5, and extend the state vector with a variable indicating the current mode (gesture) [6]:

$$\mathbf{x}' = (\mathbf{x}, \gamma), \gamma \in \{1, 2\} \qquad (7)$$

The motion model for the thumb angle is modified to include the modes:

$$p(\phi_t | \phi_{t-1}) = p(\phi_t | \gamma_t, \phi_{t-1}) p(\gamma_t | \phi_{t-1}) \qquad (8)$$

where $p(\phi_t | \gamma_t, \phi_{t-1})$ is one of the two AR processes and $p(\gamma_t | \phi_{t-1})$ is the probability for a gesture given the angle value in the last frame x_{t-1}. This last probability represents the knowledge of when the hand changes from one gesture to the other. This model will accurately follow the movement of the thumb through fast clicks with a reasonable amount of particles. Note that the representation in equation 7 means that the gesture recognition problem becomes an explicit part of the tracking process, i.e., when the tracking is correct so is the recognized gesture.

5   Likelihood Function

When the contour description of the tracked object has been created and the changes in state from frame to frame can be predicted, comparing the predicted contours with the

actual images is the next step. In the SMC tracker this is represented by the observation PDF described in this section.

This comparison is accomplished by locating edges in the image and examining how well each predicted contour corresponds to these edges. The edges are located by searching along a number of normals to the contour. Edges detected on the normals are referred to as *features*. The function that measures the correspondence is called a likelihood function and defined via a generative model.

In this section we describe a generative model inspired by the order statistic likelihood in [7]. However, where the order statistic likelihood only counts the total number of features found along the different normals, the likelihood proposed in this section will utilize individual counts of interior and exterior features, see figure 2, and we hereby obtain an IEOS (interior-exterior order statistic) likelihood function.
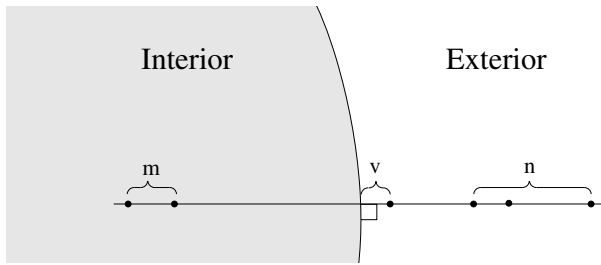
Fig. 2. Illustration of the different types of features. m is the number of features inside the hand, n is the number of features in the background, and v is the position of the feature nearest to the contour.

Both exterior and interior features are assumed to be generated by two individual Poisson processes with density parameters λ and μ, respectively. As no *a priori* knowledge exist regarding local properties of the background λ is considered constant over the entire image. μ may vary between normals to model the presence of known interior features. The density for the individual normals μ_i can be learned from training data. The probability $b_L(n)$ of finding n features on a piece of a measurement line of length L placed on the background is:

$$b_L(n) = e^{(-\lambda L)} \frac{(\lambda L)^n}{n!}$$ (9)

Similarly the probability of detecting m features on a piece of a measurement line of length L lying entirely inside the object at the position of normal number i is:

$$f_{L,i}(m) = e^{(-\mu_i L)} \frac{(\mu_i L)^m}{m!}$$ (10)

The positions of the n exterior features $\{b_1, b_2, \ldots b_n\}$ or the m interior features $\{c_1, c_2, \ldots c_m\}$ are considered uniformly distributed.

The edge of the hand is assumed to produce a single feature with a position normally distributed around the center of the normal. There is a fixed probability q_0 of this feature not being detected.

The generative model for the i'th normal with length L can be defined as:

1. Draw a from the truncated Gaussian

$$\mathcal{G}(a) = \begin{cases} ce^{\left(-\frac{a^2}{2\sigma^2}\right)} & \text{, for } a \in [-L/2; L/2] \\ 0 & \text{, otherwise} \end{cases} \tag{11}$$

   where a is the distance from the correct edge (originating from the hand) to the center of the normal, σ is found during training, and c is set so that the CDF (cumulative density function) of $\mathcal{G}(a)$ integrates to one.
2. Draw d randomly from $\{\texttt{True}, \texttt{False}\}$ with probabilities $1 - q_0$ and q_0, respectively. d represents whether the edge of the object was detected or not.
3. Draw the number of interior features m randomly from $f_{L/2+a,i}(m)$, and draw their positions $\{c_1, c_2, \ldots c_m\}$ from $\mathrm{Rect}[-L/2, a]$.
4. Draw the number of exterior features n randomly from $b_{L/2-a}(n)$, and draw their positions $\{b_1, b_2, \ldots b_n\}$ from $\mathrm{Rect}[a, L/2]$.
5. If d is $\texttt{True}$:
   (a) Set v to the position of the most central feature in the set $\{c_1, c_2, \ldots c_m, a, b_1, b_2, \ldots b_n\}$.
   If d is $\texttt{False}$:
   (a) Set v to the position of the most central feature in the set $\{c_1, c_2, \ldots c_m, b_1, b_2, \ldots b_n\}$.
   (b) If $v \in \{c_1, c_2, \ldots c_m\}$:   Set m = m -1.   Otherwise: Set n = n - 1
6. Report $\{v, m, n\}$.

The derivation of the likelihood function is divided into the two cases. One where the object edge is detected ($d = \texttt{True}$) and one where it is not ($d = \texttt{False}$).

5.1   Edge Not Found ($d = \texttt{False}$)

As v is the distance to the center of the most central of the $m + n + 1$ features found, all other features must have a distance greater than or equal to v. The PDF for the position of the most central feature can not be found directly. It will be established by determining the corresponding CDF and then differentiating.

The probability of the distance from the center to a single feature being greater than or equal to y is[1]:

$$P(|v| \geq y) = (1 - y2/L) \tag{12}$$

As the positions of the features are assumed independent, the combined probability, that k features all lie at distances from the center greater than or equal to y, can be calculated as a product of the k individual probabilities:

$$P(|v| \geq y) = (1 - y2/L)^k \tag{13}$$

---

[1] For the following four equations: $y \in [0; L/2]$.

The CDF $F(y)$ for the position of the most central of k features from a uniform distribution can be found from equation 13:

$$F(y) = P(|v| \leq y) = 1 - (1 - y2/L)^k \tag{14}$$

Differentiating 14 with regard to y yields:

$$\frac{d}{dy}F(y) = \frac{d}{dy}(1 - (1 - y2/L)^k) = \frac{2k}{L}(1 - y2/L)^{k-1} \tag{15}$$

As $|v|$ will always be in the interval $[0; L/2]$, the resulting PDF is:

$$p(v|d = \texttt{False}) = \frac{2k}{L}(1 - |v|2/L)^{k-1} \tag{16}$$

The probability of getting m interior- and n exterior features on the i'th normal, if it is centered on the border of the object, can be calculated as:

$$p_i(m, n|d = \texttt{False}) = f_{L/2,i}(m)b_{L/2}(n) \tag{17}$$

The distance from the center of the normal from the most central feature v is not used, as this feature is known to be either an interior or exterior feature and not from the edge of the object. However it is not accounted for in m or n, and it will have to be added to the right category. If the feature at v lies on the interior part of the normal, it should count as an interior feature or as an exterior feature if it lies on the outside part. That is if $v < 0$ then set $m = m + 1$ otherwise set $n = n + 1$. Adding this to equation 17 yields:

$$p_i(m, n|d = \texttt{False}, v) = \begin{cases} f_{L/2,i}(m+1)b_{L/2}(n) & \text{,if } v < 0 \\ f_{L/2,i}(m)b_{L/2}(n+1) & \text{,if } v \geq 0 \end{cases} \tag{18}$$

Equations 16 and 18 can be combined to form the likelihood that the i'th normal, centered on the border of the object, will produce $m + n + 1$ features where the most central is at position v:

$$p_i(v, m, n|d = \texttt{False}) = p_i(m, n|d = \texttt{False}, v)p(v|d = \texttt{False}) =$$
$$\begin{cases} f_{L/2,i}(m+1)b_{L/2}(n)\frac{2(m+n+1)}{L} \cdot \\ \quad (1 - |v|2/L)^{m+n} & \text{,if } v < 0 \\ f_{L/2,i}(m)b_{L/2}(n+1)\frac{2(m+n+1)}{L} \cdot \\ \quad (1 - |v|2/L)^{m+n} & \text{,if } v \geq 0 \end{cases} \tag{19}$$

The procedure for the case where the edge is found on the contour ($d = \texttt{True}$) follows a similar pattern [9] and results in:

$$p_i(v, m, n|d = \texttt{True}) = \left(\frac{2(m+n)}{L}(1 - |v|2/L)^{m+n-1}\left(1 - \int_{-|v|}^{|v|} \mathcal{G}(a)da\right) \right.$$
$$\left. + 2(1 - |v|2/L)^{m+n}\mathcal{G}(|v|) \right) f_{L/2,i}(m)b_{L/2}(n) \tag{20}$$

where v is the distance from the center of the normal to the most central feature. Combining equations 19 and 20 we obtain the IEOS likelihood function for the $i'th$ normal:

$$p_{ieos_i}(\mathbf{x}) = p_i(v, m, n) =$$
$$q_0 p_i(v, m, n | d = \texttt{False}) + (1 - q_0) p_i(v, m, n | d = \texttt{True}) \qquad (21)$$

Assuming independence of the normals, the likelihood of the entire contour corresponding to the state vector $\mathbf{x}'$ is:

$$\mathcal{P}_{ieos}(\mathbf{x}') = \prod_{i=1}^{M} p_{ieos_i}(\mathbf{x}') \qquad (22)$$

where M is the total number of normals on the contour investigated for one predicted particle (state). Equation 22 is too long to be stated in its entirety, and consequently seems to be a very costly expression to evaluate. However, this is not the case as most terms can be reduced to lookup tables [9].

5.2   Creating a Likelihood Ratio

The generative models described in the previous section form the basis for the contour likelihood functions. They can, however, also be used to derive background likelihood functions, that is, functions expressing the likelihood that a given set of features was produced by the background. Based on the generative model for the IEOS likelihood function the background likelihood for a single normal will be:

$$p_0 = b_L(f) \frac{2(f)}{L} (1 - |v|2/L)^{m+n} \qquad (23)$$

where $f = m + n + 1$. The corresponding likelihood for all M normals on the entire contour is:

$$\mathcal{B}_{ieos}(\mathbf{x}') = \prod_{i=1}^{M} b_L(f_i) \frac{2(f_i)}{L} (1 - |v_i|2/L)^{m_i+n_i} \qquad (24)$$

where $f_i = m_i + n_i + 1$. The likelihood function can now be expressed as a ratio which is more robust to noise:

$$\mathcal{R}_{ieos}(\mathbf{x}') = \frac{\mathcal{P}_{ieos}(\mathbf{x}')}{\mathcal{B}_{ieos}(\mathbf{x}')} \qquad (25)$$

5.3   Adding Color Information

In order to improve the likelihood function we learn the hue and saturation values of hands and model the colors by a Gaussian distribution. The distribution in the background is assumed to be uniform over both hue and saturation. Given these assumptions we can derive a color based ratio between the likelihood of a contour matching and the likelihood of the contour being located on a random background [9]. This color based likelihood ratio is denoted $\mathcal{R}_{color}(\mathbf{x}')$ and together with equation 25 it forms the final likelihood function used in this work:

$$\mathcal{B}_{ieosc}(\mathbf{x}') = \mathcal{R}_{ieos}(\mathbf{x}') \mathcal{R}_{color}(\mathbf{x}') \qquad (26)$$

6   Results

The HW used to test our system was the Sony Glasstron PLM-S700E HMD, a Philips ToUcam Pro USB mounted on the HMD, and a Windows PC with AMD Athlon 2100+ and 512 MB RAM. With this HW we recorded different test sequences each having different characteristics among the following: translation and rotation of the hand [slow, moderate, fast, very fast], head movement [none, slow, fast], illumination [indoor, outdoor, mixed], and background [uniform wall, wooden table, very cluttered desk].

In general the tracking of the hand and the recognition of the current gesture works well and especially the index finger is tracked reliably. In figure 3 successful tracking is illustrated for a challenging sequence with very cluttered background, fast hand movements, and additional hands entering the scene. The very fast motion of the hand and head combined with low lighting conditions cause a high level of motion blur resulting in weak edges. For a few frames with excessive motion blur, the tracker reports an erroneous position of the hand. However, track of the hand is not lost due to the stochastic nature of the SMC framework, and precise lock is regained only two frames later.
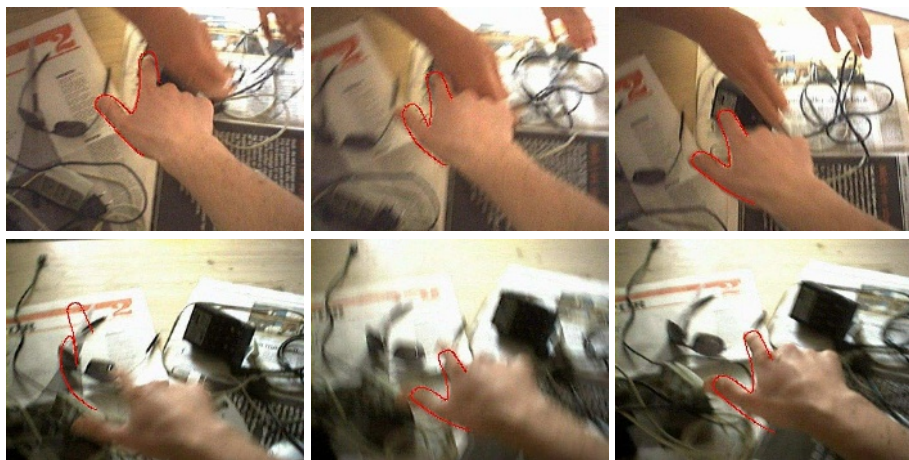
Fig. 3. Images form a test sequence. The state of the tracker at a particular frame is calculated as the weighted average of all particles and overlaid in the figure.

The speed of our system depends primarily on the number of particles required to get a robust tracking. This number in general varies in accordance with the complexity of the scene. For our test sequences the number of required particles is between 100 and 600. This corresponds to a frame rate of $100Hz$ to $12Hz$, which for this application can be considered real-time.

In order to get quantitative results we compared the estimated state (contour) with hand-segmented data and calculated the difference. In average the mean pixel error and standard deviation are around 8 and 3 for the point gesture, respectively, and around 5 and 3 for the click gesture, respectively. This is found to be usable for most interface

purposes. To stress this point we made a qualitative test where test persons were asked to control the interaction with the game pieces while playing Tic-Tack-Toe against the computer. On a uniform background the game is definitely playable. A few erroneous clicks appeared when playing on a cluttered background, especially in combination with fast hand movements. These clicks were few and could for the most parts be eliminated by a temporal filter.

Fast head motion was in general not a problem. It was observed, that during interaction the hand was kept relatively steady wrt to the head. It seems not plausible to move the head independently of the hand while pointing at something shown on the HMD.

Another very interesting issue to test is the benefits of including color information into the likelihood function. As the tracker always produces an output, i.e., a best state of the object at a particular frame, we made a sequence where the hand moves in and out of the field of view. In figure 4 the likelihoods of the best state with (left) and without (right) color information are shown. The exact values of the likelihoods are difficult to interpret as they depend on the appearance of both the object and the background. But it is evident that when the hand is not in the image (indicated by the dots) the likelihood values drop significantly when color information is used. This seems reasonable as the background is not likely to contain any hand-colors in the shape of a hand. In other words, our likelihood function provides a better signal-to-noise ratio for the system, and hence a better tracking result.
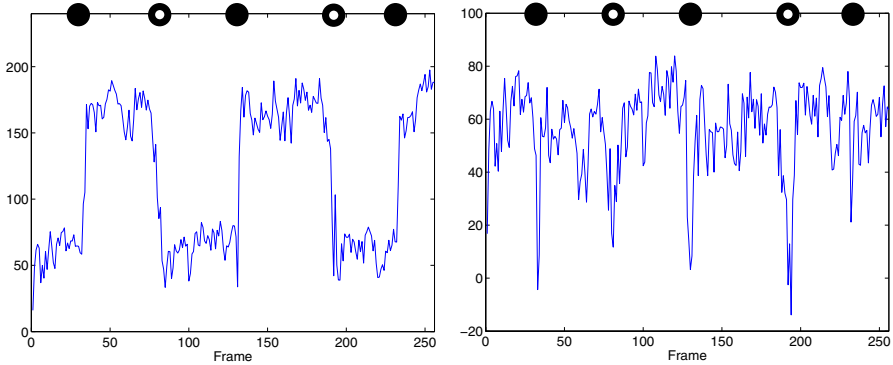
Fig. 4. The likelihood functions (left: equation 26, right: equation 22) as a function of the frame number. The black dots indicate when the hand enters the field of view while the black/white dots indicate when the hand leaves the field of view.

The clear difference in the values as a function of whether the tracker has locked on an object or not can also be used to decide when the tracker should re-initialize, i.e., use more particles or draw particles randomly from the a priori distribution. Furthermore, the steepness of the transitions in the left figure illustrates how fast the tracker regains lock when the hand reappears. In quantitative terms, the average number of frames required to regain lock is 4 for a uniform background and 7 for a cluttered background.

7   Conclusion

We have presented a non-invasive hand-gesture recognition algorithm for wearable computing. The recognized gestures are point and click gestures which are essential in virtually all interfaces where deictic information is required. Our algorithm is based on the powerful SMC tracker which can handle multiple hypotheses in the search space. In order to increase the robustness of the tracker we use ratios in the likelihood function and base it on two image cues which can complement each other: edges and skin color. The likelihood function proves to be very robust towards noise as illustrated in figure 3. Furthermore, as illustrated in figure 4 the algorithm locks on to the object very quickly and gives a clear indication of whether the hand is present in the frame or not.

The above mentioned characteristics combined with the speed of the algorithm and the user feedback allow us to conclude that we have developed a powerful deictic interface for wearable computing, and that is without requiring special lighting or markers on the hands or fingers, hence our system is a non-invasive solution.

References

1. T.J. Cham and J.M. Rehg. A Multiple Hypothesis Approach to Figure Tracking. In Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, 1999.
2. K. Choo and D.J. Fleet. People Tracking Using Hybrid Monte Carlo Filtering. In International Conference on Computer Vision, Vancouver, Canada, 2001.
3. A. Doucet, N. Freitas, and N. Gordon, editors. Sequential Monte Carlo Methods in Practice. Springer, 2001.
4. E. Foxlin and M. Harrington. Weartrack: A self-referenced head and hand tracker for wearable computers and portable vr. In International Symposium on Wearable Computing, Atlanta, Georgia, 2000.
5. M. Isard and A. Blake. CONDENSATION - conditional density propagation for visual tracking. International Journal on Computer Vision, 29(1), 1998.
6. M. Isard and A. Blake. A mixed-state CONDENSATION tracker with automatic model-switching. In International Conference on Computer Vision, Bombay, India, 1998.
7. J. MacCormick. Stochastic Algorithms for Visual Tracking: Probabilistic Modelling and Stochastic Algorithms for Visual Localisation and Tracking. Springer, 2002.
8. T.B. Moeslund and L. Nrgaard. A Brief Overview of Hand Gestures used in Wearable Human Computer Interfaces. Technical Report CVMT 03-02, AAU, Denmark, 2003.
9. L. Nrgaard. Probabilistic hand tracking for wearable gesture interfaces. Masters thesis, Lab. of Computer Vision and Media Technology, Aalborg University, Denmark, 2003.
10. W. Piekarski and B.H. Thomas. The tinmith system: demonstrating new techniques for mobile augmented reality modelling. In Australasian conference on User interfaces, 2002.
11. B. Rhodes. A brief history of wearable computing. www.media.mit.edu/wearables/lizzy/timeline.html.
12. T. Starner, J. Auxier, D. Ashbrook, and M. Gandy. The gesture pendant: A self-illuminating, wearable, infrared computer vision system for home automation control and medical monitoring. In International Symposium on Wearable Computing, Atlanta, Georgia, 2000.
13. K. Tsukada and M. Yasumura. Ubi-Finger: Gesture Input Device for Mobile Use. In Asia Pacific Conference on Computer Human Interaction, Beijing, China, 2002.
14. N. Ukita, Y. Kono, and M. Kidode. Wearable vision interfaces: towards wearable information playing in daily life. In Workshop on Advanced Computing and Communicating Techniques for Wearable Information Playing, Nara, Japan, 2002.

Gesture Recognition
Using Image Comparison Methods

Philippe Dreuw, Daniel Keysers, Thomas Deselaers, and Hermann Ney

Lehrstuhl für Informatik VI, Computer Science Department,
RWTH Aachen University, D-52056 Aachen, Germany
{dreuw, keysers, deselaers, ney}@informatik.rwth-aachen.de

Abstract. We introduce the use of appearance-based features, and tangent distance or the image distortion model to account for image variability within the hidden Markov model emission probabilities to recognize gestures. No tracking, segmentation of the hand or shape models have to be defined. The distance measures also perform well for template matching classifiers. We obtain promising first results on a new database with the German finger-spelling alphabet. This newly recorded database is freely available for further research.

1 Introduction

Work in the field of vision-based gesture recognition usually first segments parts of the input images, for example the hand, and then uses features calculated from this segmented input like shape or motion. Problems with this approach are tracking, occlusion, lighting or clothing constraints. Results in the field of object recognition in images suggest that this intermediate segmentation step is not necessary and even hindering, as e.g. segmentation or tracking is never perfect. The question addressed in our research is if appearance based features are competitive for gesture recognition and if we can use similar models of image variability as in object recognition. We have integrated distance measures known from image and optical character recognition (e.g. being invariant against affine transformations) into the hidden Markov model classifiers.

Most of the common systems [2, 8, 9, 10] assume a constant environment, e.g. persons wearing non-skin-colored clothes with long sleeves and a fixed camera position under constant lighting conditions. The presented systems are often highly person-dependent and the gestures used exhibit great differences to be easily recognizable. We aim at overcoming these shortcomings with this work.

2 Appearance-Based Features for Gesture Recognition

In appearance-based approachs the image itself and simple transformations (filtering, scaling, etc.) of the image are usually used as features. In this paper, we denote an original image X in a sequence at time $t = 1, ..., T$ by X_t, and the pixel value at the position (x, y) by $X_t(x, y)$.
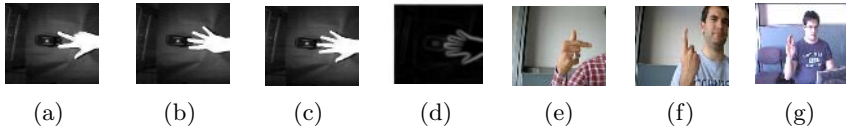
Fig. 1. Infrared images of the gesture "Five". (a)-(d): original and spatial derivatives image features. (e)-(g) are examples of the i6-Gesture database.

When working, for example, with gray valued images (e.g. infrared images like in Fig. 1(c)), *original images* or their *spatial derivatives* can be used as features. *Skin probability images* have been created according to their skin probability maps [5]. Other features have been analyzed in [3].

3   Hidden Markov Models

The ability of hidden Markov models (HMM) to compensate time and amplitude variations has been proven for speech recognition, gesture recognition, sign language recognition and human actions [4, 8, 9, 10]. In particular we focus on distance measures being invariant against slight affine transformations or distortions. The idea of a HMM is to represent a signal by a state of a stochastic finite state machine. A more detailed description can be found in [4].

In each state s of an HMM, a distance is calculated. We assume pooled variances over all classes and states, i.e. we use $\sigma_{sdk} = \sigma_d$. The negative logarithm of $p(X|s)$ can be interpreted as a distance $d(p(X|s))$ and is used as emission score:

$$-\log(p(X|s)) = \frac{1}{2}\left(\sum_{d=1}^{D}\left(\underbrace{\left(\frac{X_d - \mu_{sd}}{\sigma_d}\right)^2}_{\text{distance}} + \underbrace{\log(2\pi\sigma_d^2)}_{\text{normalization factor}} \right)\right)$$

When working with image sequences, we calculate a distance between two images, e.g. we compare the current observation image X_t (or any transformed image $\widetilde{X}_t$) with the mean image μ_s at this state. Simply comparing the pixel values is quite often used in object recognition but different methods have been proposed to do this.

Tangent Distance. Because the Euclidian distance does not account for affine transformations such as scaling, translation and rotation, the tangent distance (TD), as described in [7], is one approach to incorporate invariance with respect to certain transformations into a classification system. Here, invariant means that image transformations that do not change the class of the image should not have a large impact on the distance between the images. Patterns that all lie in the same subspace can therefore be represented by one prototype and the corresponding tangent vectors. Thus, the TD between the original image and any of the transformations is zero, while the Euclidean distance is significantly greater than zero.

Image Distortion Model. The image distortion model [6] is a method which allows for small local deformations of an image. Each pixel is aligned to the pixel with the smallest squared distance from its neighborhood. These squared distances are summed up for the complete image to get the global distance. This method can be improved by enhancing the pixel distance to compare sub images instead of single pixels only. Further improvement is achieved by using spatial derivatives instead of the pixel values directly.

4   Databases

LTI-Gesture Database. The LTI-Gesture database was created at the Chair of Technical Computer Science at the RWTH Aachen [1]. It contains 14 dynamic gestures, 140 training and 140 testing sequences. An error rate of 4.3% was achieved on this database. Fig. 1(c) shows an example of a gesture.

i6-Gesture Database. We recorded a new database of fingerspelling letters of German Sign Language. Our database is freely available on our website[1]. The database contains 35 gestures and consists of 700 training and 700 test sequences. 20 different persons were recorded under non-uniform daylight lighting conditions, without any restrictions on the clothing while gesturing. The gestures were recorded by one webcam (320x240 at 25 fps) and one camcorder (352x288 at 25 fps), from two different points of view. Fig. 1(e)-Fig. 1(g) show some examples of different gestures. More information is available on our website.

5   Results

In [1], an error rate of 4.3% has been achieved using shape and motion features in combination with forearm segmentation. Using the centroid features as presented in [8], we have achieved an error rate of 14.2%, and we can conclude that these features should only be used to describe motion patterns instead of more complex hand shapes. Using original image features on the LTI-Gesture database, we have achieved an error rate of 5.7% which has been improved to 1.4% in combination with the tangent distance [3] or the image distortion model (see Tab. 1).

Table 1. Error rates [%] on the LTI-Gesture database

Features	Euclidian	Tangent	IDM
COG [8]	14.2	–	–
original	5.7	1.4	1.4
magnitude Sobel	7.1	1.4	1.4

On the i6-Gesture database, we have used only the webcam images to test our system. It is obvious that this database contains gestures of very high complexity, and that additional methods are needed for feature extraction or other distance

---

[1] http://www-i6.informatik.rwth-aachen.de/~dreuw/database.html

measures. Using a camshift tracker to extract position independent features (note that we do *not* try to segment the hand), we could improve the error rate from 87.1% to 44.0%.

Using a two-sided tangent distance we have improved the error rate to the currently best result of 35.7%, which shows the advantage of using distance measures that are invariant against small affine transformations and the possibility of recognizing gestures by appearance-based features (see Tab. 2).

Table 2. Error Rates [%] on the i6-Gesture database

Feature	Euclidian	Tangent
original thresholded by skin color prob.	87.1	-
+ camshift tracking (no segmentation)	44.0	35.7

6   Conclusion

At this point, some questions still remain unanswered, e.g. not all distance measures and camera streams were completely analyzed on the i6-Gesture database which are expected to improve the error rate. The best achieved error rate on the i6-Gesture database is 35.7% and shows the high complexity of this database. Nevertheless, this result is promising because only a simple webcam without any restriction for the signer has been used and some signs are visually very similar, as for example the signs for "M", "N", "A", and "S".

The use of tangent distance and image distortion models as appropriate models of image variability in combination with appearance-based features has been investigated and compared to the Euclidian distance on other databases. Using these distance measures, the error rate has been reduced on all regarded databases, especially on the LTI-Gesture database. This shows the power of integrating these distance measures into the HMM emission probabilities for recognizing gestures.

References

1. S. Akyol, U. Canzler, K. Bengler, and W. Hahn. Gesture Control for Use in Automobiles. In *IAPR MVA Workshop*, Tokyo, Japan, pages 349–352, Nov. 2000.
2. R. Bowden, D. Windridge, T. Kadir, A. Zisserman, and M. Brady. A Linguistic Feature Vector for the Visual Interpretation of Sign Language. In J. M. Tomas Pajdla, editor, *ECCV*, volume 1, Prague, Czech Republic, pages 391–401, May 2004.
3. P. Dreuw. Appearance-Based Gesture Recognition. Diploma thesis, RWTH Aachen University, Aachen, Germany, Jan. 2005.
4. F. Jelinek. *Statistical Methods for Speech Recognition*. Cambridge, MA, Jan. 1998.
5. M. Jones and J. Rehg. Statistical Color Models with Application to Skin Color Detection. Technical Report CRL 98/11, Compaq Cambridge Research Lab, 1998.
6. D. Keysers, J. Dahmen, H. Ney, B. Wein, and T. Lehmann. Statistical Framework for Model-based Image Retrieval in Medical Applications. *Journal of Electronic Imaging*, 12(1):59–68, Jan. 2003.

7. D. Keysers, W. Macherey, H. Ney, and J. Dahmen. Adaptation in Statistical Pattern Recognition using Tangent Vectors. *PAMI*, 26(2):269–274, Feb. 2004.
8. G. Rigoll, A. Kosmala, and S. Eickeler. High Performance Real-Time Gesture Recognition using Hidden Markov Models. In *Int. Gesture Workshop*, volume 1371, Bielefeld, Germany, pages 69–80, Sep. 1998.
9. T. Starner, J. Weaver, and A. Pentland. Real-time ASL recognition using desk and wearable computer based video. *PAMI*, 20(12):1371–1375, Dec. 1998.
10. M. Zobl, R. Nieschulz, M. Geiger, M. Lang, and G. Rigoll. Gesture Components for Natural Interaction with In-Car Devices. In *Int. Gesture Workshop*, volume 2915 of *LNAI*, Gif-sur-Yvette, France, pages 448–459, Mar. 2004.

O.G.R.E. – Open Gestures Recognition Engine, a Platform for Gesture-Based Communication and Interaction

José Miguel Salles Dias, Pedro Nande, Nuno Barata, and André Correia

ADETTI/ISCTE, Associação para o Desenvolvimento das Telecomunicações e Técnicas de Informática, Edifício ISCTE, 1600-082 Lisboa, Portugal
{Miguel.Dias, Pedro.Nande, Nuno.Barata,
Andre.Correia}@adetti.iscte.pt
http://www.adetti.iscte.pt

Abstract. In this paper we describe O.G.R.E - Open Gestures Recognition Engine, a general purpose real time hand gesture recognition engine based on Computer Vision, able to support gesture-based communication as a modality of Human-Computer Interaction. The engine recognizes essentially, static poses of a single hand and, hand trajectory paths in simple geometrical shapes.

1   The O.G.R.E System Architecture

The O.G.R.E computing architecture (Fig. 1), requires a single video camera that captures de user's hand motion [1] and recognizes, in real time, a set of known hand poses or other types of gestures that can be used in any type of final application that may require this type of HCI modality, offering the possibility to trigger user-specified actions, activated by different hand gestures. The system initially removes the background of captured images, eliminating irrelevant pixel information, adapting itself both to changes in the lighting conditions and to the background scenario and detecting the moving foreground. The human hand is then segmented and its contours localized (in the image and vector spaces), while being also subjected to a noise reduction algorithm. From these contours, significant image or vector-based metrics are derived, allowing a search in a pre-defined generic or personal static hand poses' library, where each pose is previously converted into a set of metric values. The engine recognizes also trajectory paths, based in calligraphic interface techniques and, staged paths, hybrid gestures composed of both static poses and hand trajectories. To facilitate and simplify the use of gestures as an HCI modality by the host applications, the engine introduces the notion of Actions, an XML description of a hierarchy structure of contexts, which limits gesture recognition to a contextualized subset of possible gestures, of a given type (static, trajectory or staged path).

The essential modules of the architecture are the following:

Background subtraction (Fig. 2)**:** This is applied prior to any subsequent processing. It consists of a calibration period during which maximum and minimum per-pixel values in the YCrCb domain are stored and updated. Subsequently, foreground
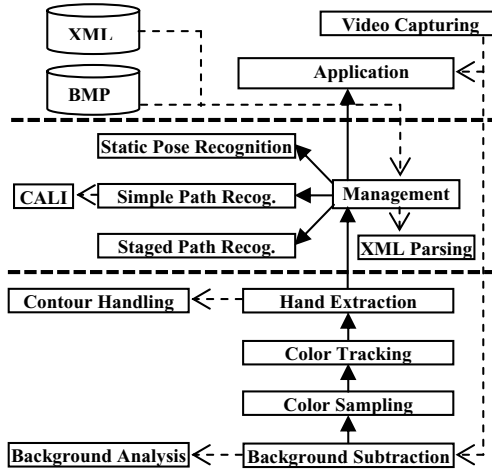
Fig. 1. OGRE system architecture

Fig. 2. Subtracting the background. From left to right: The original background; A user in the foreground; Resulting background subtraction mask; Subtracted background.

Fig. 3. Extracting the hand contour. From left to right: the CAMSHIFT algorithm histogram back projection result; Result after YCrCb re-sampling; Result after morphological smoothing; Result after contour vectorisation and mixture with the foreground image.

classification occurs, based on simple comparison between actual frame pixels' YCrCb values and the stored background model, since it is assumed that variations of actual frame pixels' YCrCb values below the stored minimum or above the stored maximum, classify these as foreground pixels.

Background Analysis: This module is responsible for background deterioration detection. It has been observed that the background subtraction algorithm used is not resilient to environmental changes, such as light fading, scene decorative objects replacement and camera positioning instability.

Color Tracking: Hue values obtained in the previous YCrCb color sampling phase, feed the CAMSHIFT (Continuously Adaptive Mean Shift) algorithm [2], which is then applied to the current captured image. The CAMSHIFT algorithm computes a histogram back projection binary image. The algorithm works in the sub-sampled chrominance domain, representing areas of a specified tonality (the hand tonality, in our case, which is a parameter of the system). It also reduces the hand pose searching area to the largest connected component representing the user's hand hue. Hand tracking is therefore guaranteed in the following frames (Fig. 3).

Hand Extraction: The CAMSHIFT algorithm identifies the largest connected component's bounding box in the histogram back projection image, but it often covers an insufficient area of it's whole. A recursive algorithm is applied to determine the real bounding box, simply enlarging its sides by small amounts and checking if it is already large enough to cover the object's area. The resulting binary image is then used as a mask for YCrCb color space re-sampling, applied to the initial image with the background removed. Luminance and chrominances are sampled at different resolutions in order to achieve the best possible contour detail. A smoothing morphological operation is then applied with an adequate structural filtering element for noise reduction. This element's dimensions can be of 5x5, 7x7, 9x9 or 11x11, depending in the estimated silhouette dimension.

Contour Handling: The extracted hand contour is converted to the vector form and, if necessary, a polygonal approximation sensitive to finger curvature is applied. This approximation is based on the best fit ellipse mathematical approach, as to obtain a measure of the curvature of a given set of points, proportional to the ellipse eccentricity. This module offers contour handling operations as to achieve an appropriate hand silhouette representation.

XML Parsing: This module is responsible for configuration, gestures and actions definitions parsing, as defined in a XML file.

Management: This module is the engine's core "intelligence". It analyses specific action context and redirects gesture recognition into the adequate system module: Static Pose Recognition, Simple Path Recognition or Staged Path Recognition.

Static Hand Pose Recognition: The extracted hand silhouette is compared against a library of silhouettes templates or a library of silhouettes signatures (depending in the method), using one of the below listed algorithms. In each case, proper formats for hand contour are used, since either we are dealing with image-based or vector-based contour analysis:

- **Image Based Analysis:** (1) Template Matching: Based in the convolution between two images at several scales in order to find a given template; (2) Discrete Cosine Transform Analysis: A scale and rotation independent domain transformation in the frequency domain.
- **Contour Based Analysis:** (3) Hu Moments: These are a set of shape characteristic invariant metrics that can be useful for shape classification; (4) Pair-wise Geometrical Histogram (PGH): This method, computes the Histogram of distances and angles between the contour polygon's edges, which provides us with a unique contour signature; (5) Simple Shape Descriptors (SSD): Combined simple

geometrical metrics, which helps describing shapes; (6) PGH-SSD Hybrid: This method corresponds to the author's effort in combining PGH and SSD advantages; (7) $C_{AL}I$ [3]: This is an open source software library, based in Fuzzy Logic, used normally to recognize sketched shapes in the context of calligraphic interfaces. The technique may bring advantage in the recognition of static hand poses, by introducing a probabilistic methodology in the recognition technique. It is based in a set of geometric measures and simple shape descriptors, such as the convex hull, the surrounding triangle and quadrangle of larger area and the surrounding rectangle of smaller area and is invariant to rigid body transformations (scale, rotation and translation).

2   Conclusions and Future Work

In this paper, we have described the different architectural modules of a real time hand gesture recognition engine based on computer-vision. The system is configured with XML specifications that describe the type of gesture to be recognized in a given context: Static Hand Poses, Simple Hand Paths or Staged Hand Paths. The system was evaluated with an experiment were a user was issuing static hand poses of Portuguese Sign Language, to assess the robustness of various algorithmic alternatives to handle with the sub-problem of shape recognition, present in the hand pose understanding process. Our results have shown that the Pair-wise Geometrical Histogram and Template Matching methods, are the most effective in relation to the average symbol recognition rate metric, reaching the average recognition rate of, respectively 58.1% and 53.6 %, for the case of the own user library of symbols. If the test is only made with highly non-correlated symbols, that metric can rise up to 90%. Taking this result into account, an application dedicated to users with hearing impairments, interacting with home appliances, was set-up, using a restricted set of static hand poses taken from the Portuguese Sign Language signs. The results were highly successful in determining the usefulness of static hand gestures in simple (but general) person-machine HCI tasks. O.G.R.E, will be soon available under GPL licensing. As a natural continuation of our work, we aim at bi-manual gesture recognition and hand feature extraction for finger recognition and occlusion treatment.

References

1. Y. Wu and T. Huang, Vision-based gesture recognition: A review, GW 1999, LNAI 1739, (1999).
2. Bradski, G., R. "Computer vision face tracking for use in a perceptual user interface". Intel Technology Journal. Microcomputer Research Lab, Santa Clara, CA, Intel Corporation (1998)
3. M. J. Fonseca and J. A. Jorge. CALI: A Software Library for Calligraphic Interfaces. http://immi.inesc.pt/projects/cali/ (2005)

Finding Motion Primitives in Human Body Gestures

Lars Reng, Thomas B. Moeslund, and Erik Granum

Laboratory of Computer Vision and Media Technology,
Aalborg University, Denmark

Abstract. In the last decade speech processing has been applied in commercially available products. One of the key reasons for its success is the identification and use of an underlying set of generic symbols (phonemes) constituting all speech. In this work we follow the same approach, but for the problem of human body gestures. That is, the topic of this paper is how to define a framework for automatically finding primitives for human body gestures. This is done by considering a gesture as a trajectory and then searching for points where the density of the training data is high. The trajectories are re-sampled to enable a direct comparison between the samples of each trajectory, and enable time invariant comparisons. This work demonstrates and tests the primitive's ability to reconstruct sampled trajectories. Promising test results are shown for samples from different test persons performing gestures from a small one armed gesture set.

1 Introduction

In the last decade speech synthesis and speech recognition have transferred from only being research topics into core technologies in commercially available products. One of the key reasons for this transfer is the identification and use of an underlying set of generic symbols constituting all speech, the phonemes. Phonemes are basically small sound samples that put together in the correct order can generate all the words in a particular language, for example English.

It is widely accepted that more than half of the information transmitted in a human-human interaction is done by other means than speech, and that the human body language is responsible for most of this information. Furthermore, for better human-computer interfaces to be build the computer might need to be equipped with the ability to understand the human body language [15]. Since automatic recognition of human body language is a desired ability research has been conducted in this area. Much of this research is based on defining a subset of the human body language, normally denoted "actions", and then building a classifier based on some kind of learning scheme applied to some training data. The result of the training is a sequence of values in some state-space for each action. The different learnt sequences are compared to the input data during run-time and a classification is carried out.

In some systems, however, a different approach is followed[1]. This approach is based on the idea that an action can be represented by a set of shorter (in terms of time duration) primitives. These primitives take different names such as movemes [4], atomic

---

[1] These approaches are sometimes motivated directly by the notion of finding "phonemes" in the human body language.

movements [5], activities [2], behaviors [12, 17], snippets [9], dynamic instants [16], states [3], and examplars [14].

Besides the different names used to describe the notion of motion primitives, the approaches also differ in another way, namely whether a primitive is dependent or independent on time. The approaches based on independence find their inspiration in key-frame animation. Key-frame animation is based on the idea that animating an articulated object in a time sequence is a matter of defining the configurations for a number of distinct frames (key-frames) and then interpolate all in-between frames using e.g., inverse kinematics. Mapping this concept to the problem of recognizing human body language converts the problem to a matter of recognizing a number of single configurations and ignoring all in-between configurations. This concept is sound but introduces a number of problems including the problem of defining which configurations (or key-frames) that best represent an action.

In the work by Rao *et al.* [16] the problem of recognizing dynamic hand gestures is addressed. They track a hand over time and hereby generate a trajectory in 3D space (x- and y-position, and time). They search the trajectory for significant changes, denoted dynamic instants, which are defined as instants with a high curvature. In the work by Jordi [8] the problem of finding key-frames for cyclic actions, like walking and running, is addressed. They capture the joint angles using an optical motion capture system and compactly represent a time sequence of such data using a point distribution model. Since the actions are cyclic they argue that the likelihood of a configuration being part of an action can be measured as the Mahalanobis distance to the mean. The key-frames are then defined as configurations where the Mahalanobis distance locally is maximum, i.e., key-frames are the least likely configurations!

The alternative to the key-frame approach is to represent the entire trajectory (one action), but doing so using a number of smaller sub-trajectories. That is, the entire trajectory through a state space is represented as opposed to only representing a number of single points. Several problems are associated with this approach, for example, how to define the length of the sub-trajectories. If too long then the primitives will not be generic. If too short the compactness of the representation is lost. In the work by Hodgins *et al.* [7] different approaches to find such sub-trajectories for full body motion are compared, and show promising results. Their comparison of three different approaches finds Probabilistic PCA as a very efficient tool for finding transitions between different behaviours.

In the work by Howe *et al.* [9] the problem of capturing the 3D motion of a human using only one camera is addressed. The main body parts are tracked in 2D and compared to learned motion patterns in order to handle the inherent ambiguities when inferring 3D configurations from 2D data. The learned motion patterns are denoted "snippets" and consist of 11 consecutive configurations. These are learned by grouping similar motion patterns in the training data. In the work by Bettinger *et al.* [1] the problem of modeling how the appearance of a face changes over time is addressed. They use an active appearance model to represent the shape and texture of a face, i.e., one point in their state-space corresponds to one instant of the shape and texture. They record and annotate a number of sequences containing facial changes. Each sequence corresponds to a trajectory in their state space. The states with the high-

est densities are found and used to divide the data into sub-trajectories. These sub-trajectories are modeled by Gaussian distributions each corresponding to a temporal primitive.

The different approaches found in the literature that uses the notion of motion primitives more or lees follow the structure below.

Temporal content. Either only a single time instant define a primitive or a primitive is based on a consecutive number of temporal instants.

Motion capture. In order to find the primitives the motion data needs to be captured. This could for example be done by an optical system or electromagnetic sensors.

Data representation. What is measured by the motion capture system is normally the 3D position of the different body parts. These measurements are often represented used normalized angles. Furthermore, the velocity and acceleration might also be considered.

Preprocessing. The captured data can have a very high dimensionality and can therefore be represented more compactly using, e.g., PCA. Furthermore, the data might be noisy and is therefore often filtered before further processing.

Primitives. It needs to be decided how to define a primitive. Often this is done via a criteria function which local minima/maxima defines the primitives.

Application. The chosen method needs to be evaluated. This can be with respect to the number of primitives versus the recognition rate, but it can also be a comparison between the original data and data synthesized using the primitives.

Our long term goal is to find a set of generic primitives that will enable us to describe all (meaningful) gestures conducted by the upper body of a human. Our approach is to investigate different data representations together with different criteria functions. We seek to find primitives for both recognition and synthesis, and evaluate the relationship between the two.

This particular paper presents the initial work towards our goal and the focus of the paper is to obtain experiences with all the topics listed above. Concretely we define a number of one-armed gestures and for each gesture we evaluate a method used to find primitives. The criteria function is based on a combination of two concepts, namely the curvature and density of a trajectory.

The paper is structured as follows. In section 2 the gesture data and the applied motion capture technique are presented. In section 3 we describe how the data is normalized. In section 4 the concept behind the primitives is given. In section 5 we present the density measure used in the criteria function, and in section 6 we combine this with a distance measure and defined how the criteria function is evaluated in order to select the primitives. In section 7 the test results are presented and in section 8 a conclusion is given.

2   The Gesture Data

The gestures we are working with are inspired by the work of [13] where a set of hand gestures are defined. The gestures in [13] are primarily two-hand gestures, but we simplify the setup to one-hand gestures in order to minimize the complexity and focus
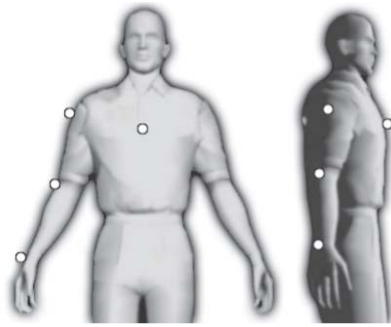
Fig. 1. Placement of sensors. The figure is adapted from [11].

on the primitives. Some of the gestures were exchanged with other more constructive ones. The final set of gestures are, as a result of this, all command gestures which can be conducted by the use of only one arm. The gestures are listed below.

Stop: Hand is moved up in front of the shoulder, and then forward (with a blocking attitude), and then lowered down.

Point forward: A stretched arm is raised to a horizontal position pointing forward, and then lowered down.

Point right: A stretched arm is raised to a horizontal position pointing right, and then lowered down.

Move closer: A stretched arm is raised to a horizontal position pointing forward while the palm is pointing upwards. The hand is then drawn to the chest, and lowered down.

Move away: Hand is moved up in front of the shoulder while elbow is lifted high, and the hand is then moved forward while pointing down. The arm is then lowered down.

Move right: Right hand is moved up in front of the left shoulder. the arm is then stretched while moved all the way to the right, and then lowered down.

Move left: Same movement as *Move right* but backwards.

Raise hand: Hand raised to a position high over the head, and then lowered down.

Each gesture is carried out a number of times by a number of different subjects, in order to have both data for inter-person comparisons, and comparable data for each gesture by several different subjects.

The gestures are captured using a magnetic tracking system with four sensors: one at the wrist, one at the elbow, one at the shoulder, and one at the torso (for reference), as shown in figure 1. The hardware used is the Polhemus FastTrac [10] which gives a maximum sampling rate of $25Hz$, when using all four sensors. In order to normalize the data and make it invariant to body size, all the collected 3-dimensional position data is converted to a time sequence of four Euler angles: three at the shoulder and one at the elbow. Besides normalizing the data, this transformation also decreases the dimensionality of the data from 12 to only 4 dimensions.

3   Normalizing the Data

In order to compare the different sequences they each need to be normalized. The goal is to normalize all the gesture trajectories so each position on a trajectory can be described by one variable t, where $t \in [0; 1]$.

The first step is to determine approximately where the gestures' endpoints are. In this experiment we have chosen to do so by defining a gesture set where all gestures are considered to both start and stop when the arm is hanging relaxed from the shoulder. A velocity threshold ensures that the small movements done between gestures is added to neither, and simplifies the separation of the individual gestures.

The trajectories are therefore homogeneously re-sampled in order to enable time invariant comparisons. This is done by interpolating each gesture, in the 4D Euler-space, by use of a standard cubic spline function. The time and velocity information is, however, still available from parameters in the new sample points, even though this is not used in this work. The homogeneously re-sampling allows for a calculation of the statistics for each gesture *and* at each sample point. Concretely, for each gesture we calculate the mean and covariance for each sample point, i.e., each instant of t. This gives the average trajectory for one gesture along with the uncertainties along the trajectory represented by a series of covariant matrices, see figure 2.
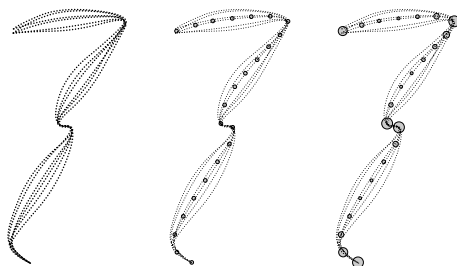
Fig. 2. Six example trajectories for a fictive gesture. Left: Input after cubic spline interpolation. Middle: Input including the position of the mean points. Right: The sizes of the mean points indicate the density of the curves.

4   Defining Primitives of Human Gestures

This section gives an intuitive description of which criteria define a good primitive candidate. In order to find the primitives we apply the following reasoning. A primitive is a particular configuration of the arm, i.e., of the four Euler angles. For a configuration to qualify as a good primitive candidate the configuration must appear in all the training data, at approximately the same time. For such a configuration to exist, all the training data must vary very little at this point in space and time, which will result in a very high density of training trajectories at this position in space. The density of a particular configuration measures how close the original sequences passed this configuration. The closer they passed the higher the density, which corresponds to

a good candidate. The logic behind this is very simple; only at the points where we have selected a primitive can we be sure that our new interpolated curve will parse directly though. Even though this paper does not go into detail with the recognition part, the main reasons for selecting primitives where the density is high is, that it makes good sense to compare an unknown curve to our known interpolated curve, at exactly the points where all the training data trajectories laid closest, see figure 2. However, just selecting the n points with the highest density will result in very inefficient primitives, since one primitive is enough to direct the interpolated curve through this area. So selecting primitives in places where the curve already passes by, will offer little to the reconstruction of the original curve. In the next two sections we describe how we calculate the density measure, and how this is used to select our primitives.

5   Measuring the Density

In section 3 the points constituting each trajectory were normalized so that the trajectories for different test subjects can be compared. That is, each trajectory was re-sampled so that they each consist of the same amount of points which are aligned. We can therefore calculate the covariance matrix for each time instant. The covariance matrices for each time instant express both how data are correlated but also how they are spread out with respect to the mean. The Mahalanobis distance expresses this relationship by defining a distance in terms of variances from a data point to the mean. It is defined as

$$r^2 = (\mathbf{x} - \boldsymbol{\mu})^T \mathbf{C}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \tag{1}$$

where $\mathbf{x}$ is a data point, $\boldsymbol{\mu}$ is the mean for this particular time instant, and $\mathbf{C}$ is the covariance matrix. If r is constant then equation 1 becomes a hyper ellipsoid in 4D space. The data points on its surface have the same variance-distance to the mean. The volume of a hyper ellipsoid with fixed Mahalanobis distance is a direct measure of the density of the data at this time instant. A big volume corresponds to a low density where the points are spread out, whereas a small volume corresponds to a high density as the same amount of data are located at a much smaller space. The volume of a hyper ellipsoid which is expressed as in equation 1 is given as [6]

$$V = \frac{\pi^2 \cdot r^4}{2} |\mathbf{C}|^{\frac{1}{2}} \tag{2}$$

where $|\mathbf{C}|$ is the determinant of the covariance matrix. We are not interested in the actual value of the volume but rather the relative volume with respect to the other time instants. Therefore equation 2 can be reduced to $V = |\mathbf{C}|^{\frac{1}{2}}$ and is illustrated in figure 2. Below we give an intuitive interpretation of this measure.

5.1   Geometrical Interpretation

Due to the inherent difficulty of illustrating in 4D we give the geometric interpretation of $|\mathbf{C}|^{\frac{1}{2}}$ in 2D and then generalize to higher dimensions.

Imagine that we have N samples in the 2D X-Y plan. For simplicity we assume that the mean of the data is the origin. The covariance matrix is given as

$$\mathbf{C} = \begin{bmatrix} c_x^2 & c_{xy}^2 \\ c_{yx}^2 & c_y^2 \end{bmatrix} \tag{3}$$

where entries are calculated as

$$c_x^2 = \frac{\Sigma x^2}{n-1} \qquad c_y^2 = \frac{\Sigma y^2}{n-1} \qquad c_{xy}^2 = c_{yx}^2 = \frac{\Sigma xy}{n-1} \tag{4}$$

We can think of $\{x_1, x_2, x_3, \cdots, x_n\}$ and $\{y_1, y_2, y_3, \cdots, y_n\}$ as two vectors in an N-dimensional space, see figure 3.
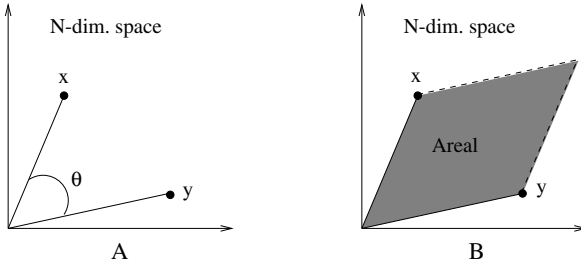
Fig. 3. A: An illustration of the 2D data interpreted as vectors in an N-dimensional space. B: The area of the parallelogram spanned by the two vectors.

The angle, θ, between the two vectors expresses the correlation between the two variables, x and y. The smaller the angle the higher correlation. The correlation is often normalized to the interval $[0, 1]$ using trigonometry, i.e., the correlation is given as $\cos(\theta)$. Relating this to the two vectors yields

$$\cos(\theta) = \frac{\mathbf{x}^T \mathbf{y}}{|\mathbf{x}||\mathbf{y}|} = \frac{\Sigma xy}{\sqrt{\Sigma x^2}\sqrt{\Sigma y^2}} \tag{5}$$

Inserting this into the covariances yields

$$c_{xy}^2 = c_{yx}^2 = \frac{\sqrt{\Sigma x^2}\sqrt{\Sigma y^2} \cos(\theta)}{n-1} \tag{6}$$

We can now calculate the determinant of the covariance matrix as

$$|\mathbf{C}| = c_x^2 \cdot c_y^2 - c_{xy}^2 \cdot c_{yx}^2 \Rightarrow \tag{7}$$

$$|\mathbf{C}| = \frac{\Sigma x^2 \Sigma y^2}{(n-1)^2} - \left(\frac{1}{n-1}\right)^2 \left(\sqrt{\Sigma x^2}\right)^2 \left(\sqrt{\Sigma y^2}\right)^2 \cos(\theta)^2 \Rightarrow \tag{8}$$

$$|\mathbf{C}| = \left(\frac{1}{n-1}\right)^2 \Sigma x^2 \Sigma y^2 \left(1 - \cos(\theta)^2\right) \Rightarrow \tag{9}$$

$$|\mathbf{C}| = c_x^2 \cdot c_y^2 \cdot \sin(\theta)^2 = \left(c_x \cdot c_y \cdot \sin(\theta)\right)^2 \tag{10}$$

This equation can be interpreted in a geometrical manner as illustrated in figure 3.B. The area, A, of the parallelogram (shaded area) is given as $A = c_x c_y \sin(\theta)$, i.e., the area depends on the standard deviation of the two variables, x and y, and the correlation between them. The higher the correlation the smaller the area. The 2D interpretation of $|\mathbf{C}|^{\frac{1}{2}}$ is the area of the parallelogram in the fourth potent. In the 3D case the geometric interpretation of the determinant of the covariance matrix is the volume of the parallelepiped spanned by three variables and their correlation. In 3D+ the geometric interpretation becomes less intuitive and is sometimes expressed as the generalization of the concept of variance.

6   Selecting the Primitives

Above we have defined and presented a method for calculating the density measure, and are now ready to include this into one criteria function that can be evaluated in order to find the primitives. The criteria function will combine the density measure with the distance between the homogeneously re-sampled mean gesture trajectory (m) and a trajectory made by interpolating the endpoints and the first selected primitives, using a standard cubic spline function (c) for each of the four Euler angles. In order to make a direct comparison, both the mean gesture trajectory and the interpolated cubic spline trajectory were given the same amount of points. This enables a calculation of the *error*-distance (δ) between the curves for each point pair. If multiplying this error distance at each point with the density (β), we can get a distance measure much similar to the Mahalanobis.

Since the four angles might not have the same dynamic ranges and more freedom to optimize future parameters is desired, the criteria function (λ) is defined as a weighted sum of error measures (α_i) for each of the four Euler angles:

$$\lambda(t) = \omega_1 \alpha_1(t) + \omega_2 \alpha_2(t) + \omega_3 \alpha_3(t) + \omega_4 \alpha_4(t) \tag{11}$$

where the four weights $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$, and the error measure:

$$\alpha_i(t) = \beta_i(t) \cdot \delta_i(t)^2, and : \delta_i(t) = \sqrt{(m_i(t) - c_i(t))^2} \tag{12}$$

Given the criteria function in equation 11 we are now faced with the problem of finding the N best primitives for a given trajectory. The most dominant primitive, χ_1 is obviously defined as : $\chi_1 = arg \max_t \lambda(t)$.

In order to find the second primitive, the first one is added to the cubic spline function (c), and the four trajectories are then recalculated, so new error distance measures can be calculated, see figure 4. This procedure can be repeated until the sum of all (λ) falls below a given threshold, or the number of primitives reaches an upper threshold.

6.1   Optimizing the Primitive's Position

Knowing that this method can, most likely, be improved; we tried to implement an optimizing step at the end of each primitive selection. A brute force test on all the test data could be used in order to find the optimal solution given a number of maximum
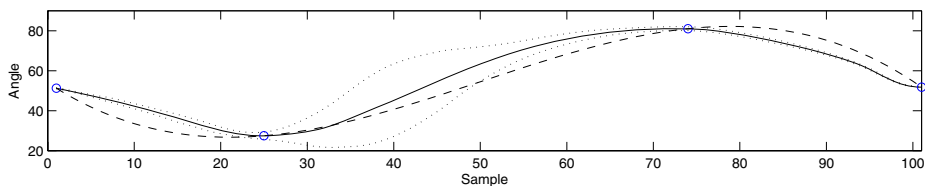
Fig. 4. Calculating the error-distance in one direction. Solid: The mean gesture trajectory. Dashed: Interpolated cubic spline. Dotted: Variance of training data. Circles: Selected primitives and endpoints.

primitives and number of samples. This is, however, very time consuming, and only valuable for the given data set, and was therefore not considered.

Instead, tests were done with another much faster method. After each new primitive was selected, all the selected primitives were tested in a position one step to each side along the mean gesture trajectory. Only if they could lower the total error sum, will they move to this position, and as long as just one primitive could be moved, all other would be tested again. This method will bring the error sum to a local minimum, but not to a guarantied global minimum.

See the following section for tests results on both previous described methods.

7 Results

The tests described in this section were made on a training data set based on the eight one arm gestures described in section 2. Three tests persons conducted each gesture no less than ten times resulting in a total of 240 gestures[2]. The evaluation of our approach consists of two tests for each action:

- Investigate how many primitives are required in order to reconstruct the original gestures.
- Evaluate the optimization step, and determine whether or not this should be used in our continuous work.

It is our belief that the only reasonable way to evaluate whether the reconstruction of a gesture is life like enough to look natural, is to have a robot or virtual human avatar performing the reconstructed gestures before a large number of test persons, and having these evaluate the result. This was however not within range of our possibilities at this point in our research. Instead, all reconstructions were evaluated by the research group from a large number of graphs such as those shown in figures 5 and 6. The graphs show the four angle spaces and error measure of the gesture *Move Left*, with two endpoints and 2,4 and 8 primitives. Figure 5 show the result of the reconstruction without the optimizing step, where as 6 were depicture the reconstruction of the exact same angle spaces, but with the optimization. The sum of the error measures for each curve pair of

---

[2] Additional 160 training gestures were made but had to be removed from the set do to extremely low signal to noise ratio.

Fig. 5. Reconstruction and error. Solid: The mean gesture trajectory. Dashed: Interpolated cubic spline. Dotted: Variance of training data. Circles: Selected primitives and endpoints. A: With 2 primitives. B: With 4 primitives. C: With 8 primitives.
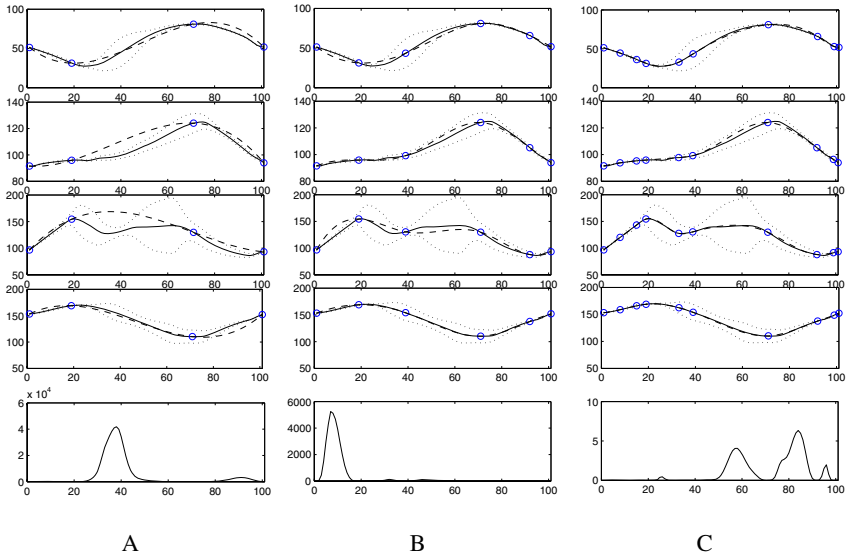

Fig. 6. Reconstruction and error (Optimizted version). Solid: The mean gesture trajectory. Dashed: Interpolated cubic spline. Dotted: Variance of training data. Circles: Selected primitives and endpoints. A: With 2 primitives. B: With 4 primitives. C: With 8 primitives.
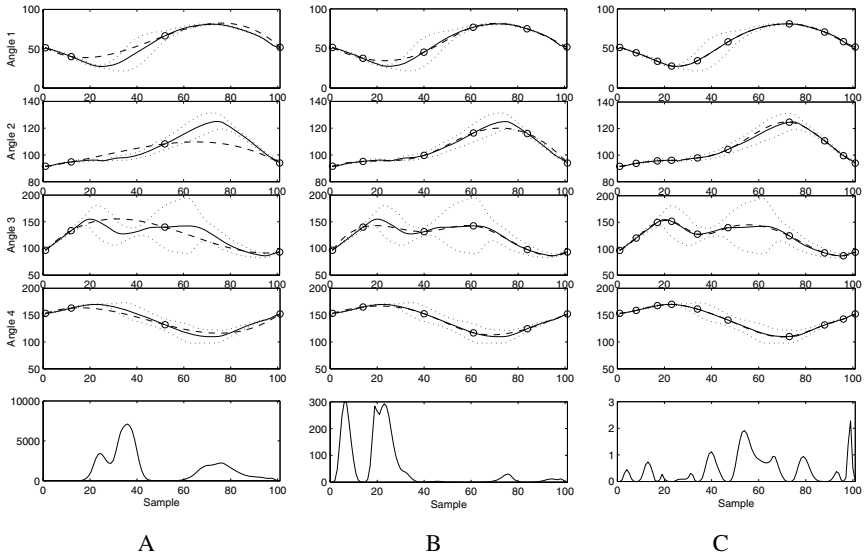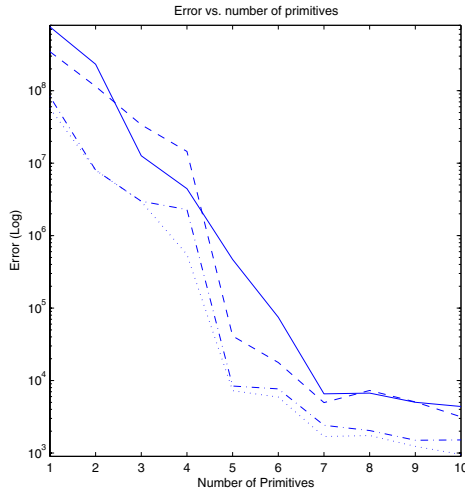
Fig. 7. Logaritmic scale of error vs. number of primitives. Solid: Reconstruction error after primitive selection without the density measure. Dashed: Reconstruction error after primitive selection with the density measure. Dash-dot: Reconstruction error after primitive selection without the density measure, but with optimization. dotted: Reconstruction error after primitive selection with the density measure and optimization.

each gesture, were collected after the reconstruction of the curves with the number of primitives ranging from 1-10. The error sums of both the optimized and none optimized version of our approach are plotted on a single logarithmic graph, shown in figure 7. The graph shows clearly that the optimized version has a lover error sum, but also that one or more of the reconstructions with four primitives were stranded in an unfortunate local minimum.

8   Conclusion

In this paper we have presented a framework for automatically finding primitives for human body gestures. A set of gestures is defined and each gesture is recorded a number of times using a commercial motion capture system. The gestures are represented using Euler angles and normalized. The normalization allows for calculation of the mean trajectory for each gesture along with the covariance of each point of the mean trajectories. For each gesture a number of primitives are found automatically. This is done by comparing the mean trajectories and cubic spline interpolated reconstructed trajectory by use of a error measurement based on density.

Our framework were implemented in two slightly different versions, were the slower proved to be superior, as it often is. Taken into consideration that our training data were very noisy, and the presented work is part of an ongoing research, we find the current results very promising, and will continue our work in this direction. We feel that the density measure have been proven as a factor that must be considered in

this line of work. Its is still hard to say exactly how many primitives are needed to get a natural reconstruction of a given gesture. But our tests indicate that somewhere between five and ten should be sufficient. It is obvious that other kind of curve-reconstruction techniques should result in much better reconstruction. But since the key-frames are to be used for recognition as well, it is important to have our key-frames at the points where the density is highest.

References

1. F. Bettinger and T.F. Cootes. A Model of Facial Behaviour. In *IEEE International Conference on Automatic Face and Gesture Recognition*, Seoul, Korea, May 17 - 19 2004.
2. A.F. Bobick. Movemnet, Activity, and Action: The Role of Knowledge in the Perception of Motion. In *Workshop on Knowledge-based Vision in Man and Machine*, London, England, Feb 1997.
3. A.F. Bobick and J. Davis. A Statebased Approach to the Representation and Recognition of GEstures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(12), 1997.
4. C. Bregler. Learning and Recognizing Human Dynamics in Video Sequences. In *Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 1997.
5. L. Campbell and A.F. Bobick. Recognition of Human Body Motion Using Phase Space Constraints. In *International Conference on Computer Vision*, Cambridge, Massachusetts, 1995.
6. R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley & Sons, Inc., 2 edition, 2001.
7. Hodgins et al. Segmenting Motion Capture Data into Dstinct Behaviors. *Unknown*, 2004.
8. J. Gonzalez. *Human Sequence Evaluation: The Key-Frame Approach*. PhD thesis, Universitat Autonoma de Barcelona, Barcelona, Spain, 2004.
9. N.R. Howe, M.E. Leventon, and W.T. Freeman. Bayesian Reconstruction of 3D Human Motion from Single-Camera Video. In *Advances in Neural Information Processing Systems 12*. MIT Press, 2000.
10. http://polhemus.com/. Polhemus, three-dimensional scanning, position/orientation tracking systems, eye tracking and head tracking systems., January 2005.
11. http://www.3dcrimescene.com/. Typical cold case reconstruction., January 2005.
12. O.C. Jenkins and M.J. Mataric. Deriving Action and Behavior Primitives from Human Motion Data. In *International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Sep 2002.
13. A. Just and S. Marcel. HMM and IOHMM for the Recognition of Mono- and Bi-Manual 3D Hand Gestures. In *ICPR workshop on Visual Observation of Deictic Gestures (POINTING'04)*, August 2004.
14. A. Kale, N. Cuntoor, and R. Chellappa. A Framework for Activity-Specific Human Recognition. In *International Conference on Acoustics, Speech and Signal Processing*, Orlando, Florida, May 2002.
15. T.B. Moeslund and E. Granum. A Survey of Computer Vision-Based Human Motion Capture. *Computer Vision and Image Understanding*, 81(3), 2001.
16. C. Rao, A. Yilmaz, and M. Shah. View-Invariant Representation and Recognition of Actions. *International Journal of Computer Vision*, 50(2), 2002.
17. C.R. Wren and A.P. Pentland. Understanding Purposeful Human Motion. In *International Workshop on Modeling People at ICCV'99*, Corfu, Greece, September 1999.

Gesture Analysis of Violin Bow Strokes

Nicolas H. Rasamimanana, Emmanuel Fléty, and Frédéric Bevilacqua

IRCAM, 1, Place Igor Stravinsky, 75004 Paris
{Nicolas.Rasamimanana, Emmanuel.Flety,
Frederic.Bevilacqua}@ircam.fr

Abstract. We developed an "augmented violin", i.e. an acoustic instrument with added gesture capture capabilities to control electronic processes. We report here gesture analysis we performed on three different bow strokes, *Détaché*, *Martelé* and *Spiccato*, using this augmented violin. Different features based on velocity and acceleration were considered. A linear discriminant analysis has been performed to estimate a minimum number of pertinent features necessary to model these bow stroke classes. We found that the maximum and minimum accelerations of a given stroke were efficient to parameterize the different bow stroke types, as well as differences in dynamics playing. Recognition rates were estimated using a kNN method with various training sets. We finally discuss that bow stroke recognition allows to relate the gesture data to music notation, while a bow stroke continuous parameterization can be related to continuous sound characteristics.

Keywords: Music, Gesture Analysis, Bow Strokes, Violin, Augmented Instruments.

1 Introduction

There is an increasing interest in using gestural interfaces to control digital audio processes. Despite numerous recent achievements ([12]), important ground work on gesture analysis is still necessary for the improvement of such interfaces. We are currently developing various "augmented instruments", i.e. acoustic instruments with added gesture capture capabilities. Such an approach remains remarkably fruitful for the study of gesture in music. As a matter of fact, the use of acoustic instruments in this context allows to apprehend instrumental gesture in a *a priori* defined framework, linked to both a symbolic level, the music notation, and a signal level, the acoustic instrument sound.

One of our current project concerns an "augmented violin", similar to the one developed by D. Young [13]. On a fundamental level, our goal is to build a model of the player's gestures reflecting his/her expressive intentions related to violin playing techniques. Specifically, our aims are to establish the relationships between the captured data, bowing styles and sound characteristics. This includes the study, on a gestural level, of the variations that occur between different interpretations of a single player or between players. These studies will

lead us to the development of real-time analysis tools, enabling an *interpretation feedback*, which includes gesture recognition, and *gesture following*, i.e. the possibility to track a performance with respect to a predefined reference. We believe that both approaches are key to develop novel types of interaction between instrumentalists and computers.

We report in this paper the study of three violin bow strokes (*Détaché, Martelé* and *Spiccato*) and the evaluation of their possible recognition. The article is organized as follows. We first present a review of similar works. In section 3, we present the capture system implemented on the violin. In sections 4 and 5, we show results on the parameterization and recognition of bow stroke types. Finally, we conclude in sections 6 and 7 by a discussion of these results and their implications on future work.

2   Related Works

Our concept of "augmented instruments" is similar to the Hyperinstruments developed by T. Machover and collaborators. The idea is to use a traditional instrument and to extend its capabilities by digital means. For example, the HyperCello [7] created in 1991 was conceived as an acoustic cello with added measurements of wrist movement, bow pressure and position, and left hand fingering. More recently, D. Young extended the HyperCello to the violin with the HyperBow [13], [14].

Several other interfaces have been developed based on string instruments for artistic purposes ([6], [4], [11], and [9]). All of these works generally used the sensor signals to directly control sound treatment parameters, such as filters [11] or physical model synthesis [14]. B. Schoner [10] adopted a probabilistic approach to infer, in real time, cello sounds from the gesture input given by the HyperCello.

Very few works actually report an analysis of the signals, specifying the relationships between the data and the instrumentist's performance. Among them, C. Peiper et al. [8] used decision tree techniques to classify violin bow strokes based on motion tracking. We here pursue the approach of analyzing different types of bow strokes, and in particular we propose to estimate invariance and variability of the measured signals.

3   Hardware Design

Hardware developments were designed with the following constraints: compatibility with an acoustic violin, no significant alteration of the instrument, wireless communication, relatively inexpensive. The prototype we built and used in this study is shown on figure 1. Two types of gesture data are measured, using technology similar to the one described in [13]: bow position and bow accelerations.

First, the sensing system can measure the bow-strings contact position along two directions: between tip and frog, between bridge and finger-board. This

position is measured via capacity coupling between a resistive tape fixed along the bow and an antenna behind the bridge.

Second, acceleration is sensed thanks to two Analog Device ADXL202 placed at the bow frog. Note that such sensors are sensitive to both gravity, hence inclination, and movement acceleration (generally referred as static and dynamic accelerations). The two accelerometers are fixed to the bow nut in such a way that acceleration is measured in three dimensions: bowing direction, string direction and vertical direction.

The position data, obtained from the antenna behind the bridge, is digitized in 16 bits with a sensor acquisition system developed at IRCAM, *Ethersense* [3]. The acceleration data are sent wirelessly to a RF receiver also connected to the sensor acquisition system. The acceleration dynamic range has been measured to be of 65 dB. All the data are transmitted to Max/MSP through an ethernet connection using the Open Sound Control protocol, at a data rate of 200 Hz. The surplus weight added by the sensing system is actually 15 grams, mainly located at the frog. Although perceptively heavier, the bow is still playable according to professional violinists. A smaller and slightly lighter prototype is currently under development.
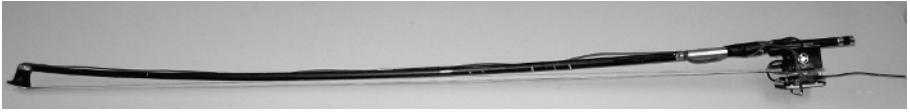
(a). Augmented violin bow
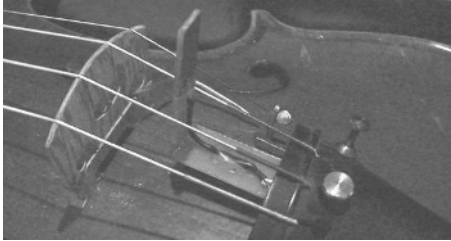
(b). Antenna behind the bridge for position measurements.
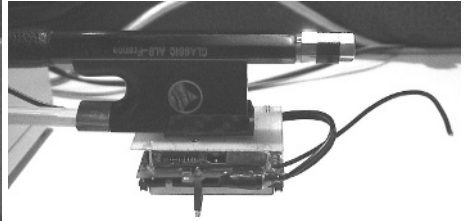
(c). The sensing system placed on the bow frog.

Fig. 1. Pictures of the augmented violin prototype

4   Gesture Analysis

We studied three standard types of bow strokes (*Détaché*, *Martelé* and *Spiccato*), by focusing the analysis on accelerometer signals in the bowing direction, which contain the essential information.

4.1   Violin Bow Strokes

Here is a brief description of these bow strokes according to [2].

In *Détaché*, the bow linearly goes from tip to frog and inversely from frog to tip. This linear movement must be adapted to the various dynamics. The bow can be used entirely or in fractions.

Martelé requires a violent gesture. The whole arm must be rapid and vigorous: a very sharp, almost percussive attack must be obtained at each extremity of the bow.

Spiccato uses the phalanges suppleness so that the bow can leave the string after each notes. It results in a light and precise sound.

4.2   Data Acquisition

We built a database from recordings of professional and amateur violinists performing scales in the three bow strokes *Détaché*, *Martelé* and *Spiccato*, at two tempi, 60 bpm and 120 bpm, and three dynamics, *pianissimo* (*pp*), *mezzo forte* (*mf*), *fortissimo* (*ff*).

In order to free the accelerometer signals from angle contributions, we asked the violinists to perform scales on one string at a time and recorded scales on every strings. This way, angle contribution is a constant offset and can be subtracted.

We chose in this study to consider individual strokes. We therefore segmented the recorded gesture data using a peak detection algorithm on the acceleration signals. The gesture database is hence constituted of executions of separate notes,
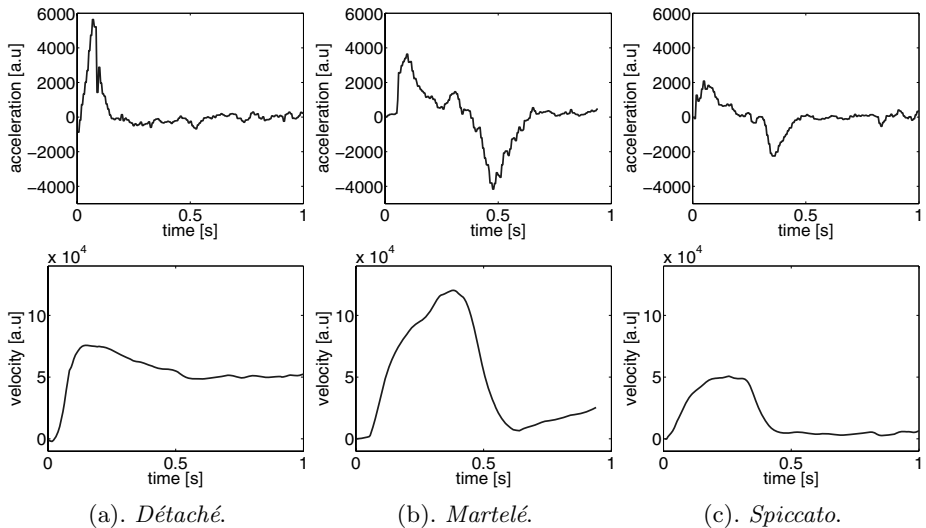

(a). *Détaché.*   (b). *Martelé.*   (c). *Spiccato.*

Fig. 2. Acceleration and velocity curves for a single note played in the three styles *Détaché*, *Martelé* and *Spiccato*. Dynamic is *mf* and tempo 60 bpm.

played in three different styles, at three dynamics, two tempi, by two different players.

Figure 2 shows an example of data for the three types of bow strokes *mf* and at 60 bpm. We can see that in *Détaché*, bow velocity remains relatively constant after the attack, unlike *Martelé* and *Spiccato*, where the bow must be slowed down. *Martelé* has typically higher absolute acceleration values compared to *Spiccato*. *Martelé* indeed requires more velocity as it is generally performed using a greater length of bow, compared to *Spiccato*, in order to achieve its typical percussive attack.

4.3   Gesture Features

Four parameters are derived from the acceleration and velocity curves to model the bow strokes: a_{max}, a_{min}, v_{max} and v_{min} (first local minimum after v_{max}), as illustrated on figure 3. Bow velocity is computed from the integration of accelerometers signals. These features correspond to a basic parameterization of the velocity curve shape. They can be computed with sufficient precision and without assuming any model for the velocity shape. They allow for the representation of *Détaché*, *Martelé* and *Spiccato* within a four dimensional space.
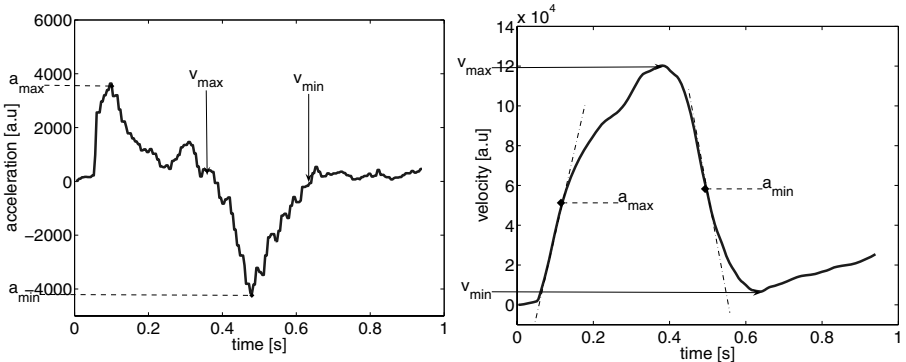
(a). Features on the Acceleration Curve.   (b). Features on the Velocity Curve.

Fig. 3. Illustration of the four features a_{max}, a_{min}, v_{max} and v_{min} (first local minimum after v_{max}) on *Martelé* acceleration and velocity curves

4.4   Gesture Space

We used Linear Discriminant Analysis (LDA), which maximizes separation between classes, to estimate the dimensionality of the parameterization. LDA on the gesture database, considering three bow strokes classes, indicates that the class scatter matrix only has two significant eigen values. Therefore, the gesture data can be clustered in a bidimensional space, with maximum in-between classes distance.

(a). Player 1.   (b). Player 2.
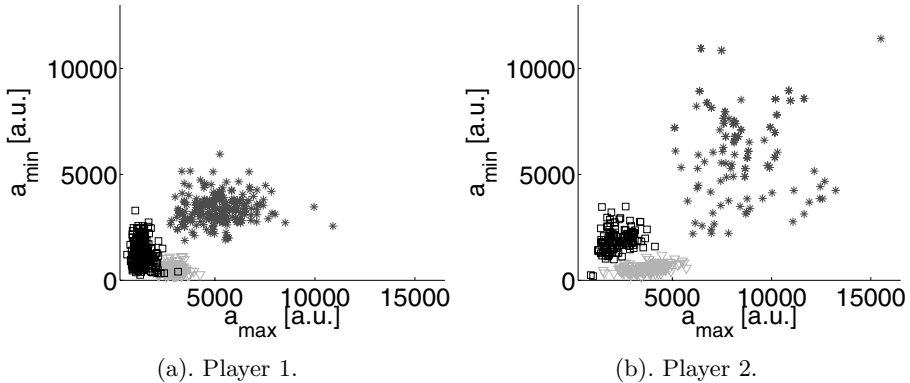
Fig. 4. Bow Strokes Feature Space (Player Detail). Each point corresponds to a single bow stroke. Fig (a) and (b) show the feature space for each player, at a same dynamic (mf) and tempo (60 bpm). Legend is *Détaché* $= \nabla$, *Martelé* $= *$, and *Spiccato* $= \square$.

We actually found that a_{max} and a_{min}, having major contributions in the eigen vectors, are the two most consistent parameters to model bow strokes, as illustrated in figures 4 and 5. As shown on figures 4(a) and 4(b), for a given dynamic, each bow stroke type forms a separate cluster. Moreover, the disposition of these clusters is similar for both players.

Figure 5(a) illustrates the case where different dynamics are considered. The basic clustering structure remains even if overlap occurs. Nevertheless, for each bow stroke types, sub-structure clustering can be observed as detailed in figures 5(b), 5(c) and 5(d). Precisely, each cluster is composed of three sub-clusters, one for each dynamic variations (*pp*, *mf*, *ff*). *Fortissimo* always corresponds to the highest a_{max} and a_{min} values.

5   Gesture Recognition

We further evaluate the ability of recognizing bow stroke using kNN with a_{max} and a_{min}. Three different test scenarios were chosen.

First, we defined three classes, corresponding to the three types of bow strokes. The whole database, i.e. mixing two players, three dynamics and two tempi, is

Table 1. kNN recognition results (Test scenario 1). Database is mixing 2 players, 3 nuances and 2 tempi. Three classes considered.

Test\Ref	Détaché	Martelé	Spiccato
Détaché	**96.7%**	1.3%	2.0%
Martelé	1.0%	**85.8%**	13.2%
Spiccato	6.0%	5.0%	**89.0%**
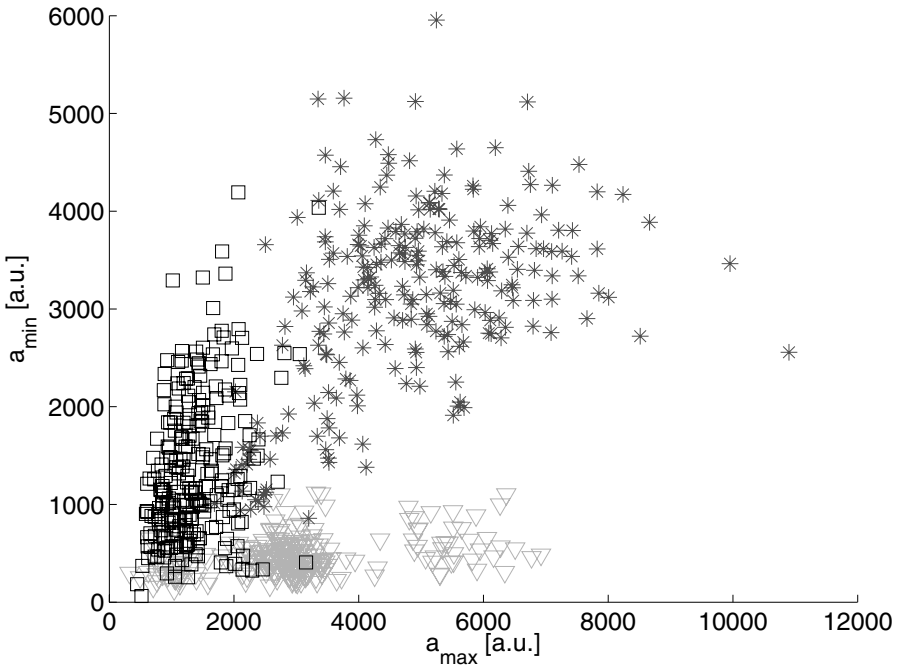
(a). Bow stroke feature space mixing three dynamics for one player and one tempo (60 bpm). Legend is *Détaché* = ∇, *Martelé* = $*$, and *Spiccato* = $\square$.
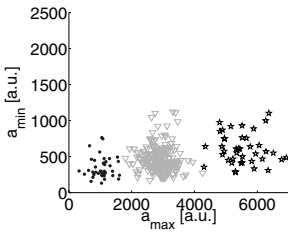
(b). *Détaché* cluster:
(pp= $\bullet$, mf= ∇, ff= $\star$).
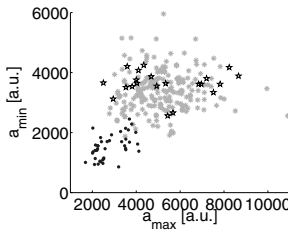
(c). *Martelé* cluster:
(pp= $\bullet$, mf= $*$, ff= $\star$).
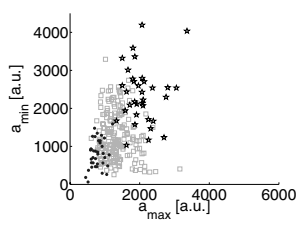
(d). *Spiccato* cluster:
(pp= $\bullet$, mf= $\square$, ff= $\star$).

Fig. 5. Bow Strokes Feature Space (Dynamic Detail). Each point corresponds to a single bow stroke. Fig (a) plots all the features points for one player, at one tempo and at three dynamics. The three bow strokes appear in clusters. Fig (b), (c) and (d) show the detail for each bow stroke cluster: three sub-clusters corresponding to the three dynamics can be seen.

randomly divided into two parts (one-fourth and three-fourths). The quarter of the database, i.e. 320 points, serves as a reference and the remaining three quarter, i.e 1000 points, is used to evaluate the recognition rate. For each test

point, vote is done according to the most represented type of bow stroke in the 10 nearest neighbors. Table 1 shows the recognition percentage for this first setup.

In the second scenario, we considered the same three classes but cross-tested the players data: one served as reference for the other. The results are reported in table 2.

Table 2. kNN recognition results (Test scenario 2). Database is mixing 1 player (*Pl1*), 1 nuance (*mf*) and 1 tempo (60bpm). Test points from other player (*Pl2*), same nuance and tempo. Three classes considered.

Ref		*Pl1*		
Test \		*Det*	*Mar*	*Spi*
	Det	**100.0%**	0.0%	0.0%
Pl2	*Mar*	0.0%	**100.0%**	0.0%
	Spi	6.3%	25.0%	**68.7%**

In the third scenario, we considered each variation of dynamics as a separate class. Thus, nine reference classes, i.e. three types of bow strokes times three nuances for a single player, are tested. This time, two-thirds of the database are used as a reference where each of the nine classes is represented. Table 3 shows the recognition results. For each line, first column is the class of the tested points and the other columns give the percentages of recognition for the nine classes.

Table 3. kNN recognition results (Test scenario 3). Database is mixing 1 player (*Pl1*), 3 nuances, 1 tempo (60bpm). Nine classes considered (3 bow strokes x 3 nuances).

Ref		*pp*			*mf*			*ff*		
Test \		*Det*	*Mar*	*Spi*	*Det*	*Mar*	*Spi*	*Det*	*Mar*	*Spi*
	Det	**100.0 %**	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
pp	*Mar*	0.0 %	**78.6 %**	0.0 %	0.0 %	0.0 %	**7.1 %**	0.0 %	0.0 %	**14.3%**
	Spi	23.1 %	0.0 %	**76.9 %**	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
	Det	0.0 %	9.5 %	0.0 %	**90.5 %**	0.0 %	0.0 %	0.0 %	0.0 %	0.0 %
mf	*Mar*	0.0 %	0.0 %	0.0 %	0.0 %	**95.8 %**	0.0 %	4.2 %	0.0 %	0.0%
	Spi	0.0 %	35.3 %	0.0 %	0.0 %	0.0 %	**64.7 %**	0.0 %	0.0 %	0.0 %
	Det	0.0 %	0.0 %	0.0 %	6.7 %	0.0 %	0.0 %	**93.3 %**	0.0 %	0.0 %
ff	*Mar*	0.0 %	0.0 %	0.0 %	0.0 %	85.7 %	0.0 %	0.0 %	**14.3 %**	0.0%
	Spi	0.0 %	0.0 %	0.0 %	0.0 %	0.0%	**49.9 %**	0.0 %	0.0 %	**50.1%**

6   Discussion

For recognition, three test scenarios were elaborated. The first two scenarios yields high recognition rates. This shows that the three bow strokes are efficiently

characterized by the features (a_{min}, a_{max}), even mixing data of two players, different dynamics and tempi, and with a relatively low number of reference data, i.e. one-fourth of the data. Moreover, the cross-player test done in the second scenario confirms the features invariance properties. In this well defined playing situation (scales), our results thus show that the chosen features can be directly related to a music notation level.

In the third scenario, the recognition performances are reduced in some cases. Even with a high proportion of data as reference (two-thirds), confusions occur for example between *Spiccato mf* and *Martelé pp*. However, such confusions are informative as they illustrate actual similarities in bow stroke gestures, when mixing different dynamics. Precisely, from our results, the following different classes, *Spiccato mf*, *Martelé pp* and *Détaché mf*, share similar features, which was actually found to be consistent from the viewpoints of violinists. This shows the limits of recognition approaches since frontiers between classes are not always well defined perceptively.

Furthermore, points that are close in the gesture feature space (figure 5(a)) share similar sound characteristics, e.g. *Martelé pp*, *Détaché mf* and *Spiccato ff*. Consequently, it is perceptually more coherent to characterize bow strokes with a continuous parameterization, using for example a_{max} and a_{min}: such parameters can indeed be related to continuous sound characteristics and/or perceptual features of the listener. It is important to note that a continuous parameterization enables both the recognition of bowing styles and the characterization of hybrid bow strokes.

The results of the study also show that bow acceleration is a parameter of major influence to characterize the different ways of bowing. This comes in complement to acoustic studies on the violin, notably by A. Askenfelt [1] and K. Guettler [5], having already demonstrated the influence of bow acceleration values on the establishments of a Helmhotz regime. It will be interesting to relate the different bowing styles to the number of nominal periods elapsing before Helmholtz triggering occurs, as described in [5].

7   Conclusion and Perspectives

Our goal was to study three different bow strokes, *Détaché*, *Martelé* and *Spiccato*, based on gesture data. After considering basic features based on velocity and acceleration curves, we found that a_{max} and a_{min} provided a pertinent parameterization of these bow strokes. In particular, these parameters enable the recognition of bow stroke types (even in the case of two different players). When considering a higher number of classes including dynamics, we noted typical confusions, consistent with perceptual point of views of violin players and listeners. In summary, our gesture analysis was based on two complementary approaches: recognition and gesture parameterization. Recognition allows us to relate gesture data to music notation, while continuous parameterization of bow strokes could be related to continuous sound characteristics. The

detailed relationship between gesture data and sound parameters will be the object of a future study. Moreover, we will investigate other type of parameterizations of the velocity and acceleration that should account for finer characterization of bow strokes. Other parameters such as bow force on strings, pointed by acoustic studies as an influential parameter on sound, will also be considered.

Acknowledgments

References

1. Anders Askenfelt. Measurement of the bowing parameters in violin playing. ii: Bow-bridge distance, dynamic range, and limits of bow force. *J. Acoust. Soc. Am.*, 86(2), 1989.
2. Ami Flammer and Gilles Tordjman. *Le Violon.* J.-C. Lattès and Salabert, 1988.
3. Emmanuel Fléty, Nicolas Leroy, Jean-Christophe Ravarini, and Frédéric Bevilacqua. Versatile sensor acquisition system utilizing network technology. In *Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2004.
4. Camille Goudeseune. *Composing with parameters for synthetic instruments.* PhD thesis, University of Illinois Urbana-Champaign, 2001.
5. Knut Guettler. On the creation of the helmholtz motion in bowed strings. *Acta Acustica*, 88, 2002.
6. Charles Nichols. The vbow: Development of a virtual violin bow haptic human-computer interface. In *Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2002.
7. Joseph A. Paradiso and Neil Gershenfeld. Musical applications of electric field sensing. *Computer Music Journal*, 21(2), 1997.
8. Chad Peiper, David Warden, and Guy Garnett. An interface for real-time classification of articulations produced by violin bowing. In *Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2003.
9. Cornelius Poepel. Synthesized strings for string players. In *Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2004.
10. Bernd Schoner, Chuck Cooper, Chris Douglas, and Neil Gershenfeld. Cluster-weighted sampling for synthesis and cross-synthesis of violin family instrument. In *Proceedings of the International Computer Music Conference, ICMC*, 2000.
11. Dan Trueman and Perry R. Cook. Bossa: The deconstructed violin reconstructed. In *Proceedings of the International Computer Music Conference, ICMC*, 1999.

12. M.M. Wanderley and M. Battier, editors. *Trends in Gestural Control of Music*. Ircam, 2000.
13. Diana Young. The hyperbow controller: Real-time dynamics measurement of violin performance. In *Proceedings of the Conference on New Instruments for Musical Expression, NIME*, Dublin, Ireland, 2002.
14. Diana Young and Stefania Serafin. Playability evaluation of a virtual bowed string instrument. In *Proceedings of the Conference on New Instruments for Musical Expression, NIME*, 2003.

Finger Tracking Methods Using EyesWeb

Anne-Marie Burns[1] and Barbara Mazzarino[2]

[1] Input Devices and Music Interaction Lab, Schulich School of Music,
McGill University, 555 Sherbrooke Street West,
Montréal, Québec, Canada, H3A 1E3
anne-marie.burns@mail.mcgill.ca
http://www.music.mcgill.ca/~amburns/
[2] InfoMus Lab, DIST - University of Genova,
Viale Causa 13, I-16145, Genova, Italy
barbara.mazzarino@unige.it
http://infomus.dist.unige.it

Abstract. This paper compares different algorithms for tracking the
position of fingers in a two-dimensional environment. Four algorithms
have been implemented in EyesWeb, developed by DIST-InfoMus labo-
ratory. The three first algorithms use projection signatures, the circular
Hough transform, and geometric properties, and rely only on hand char-
acteristics to locate the finger. The fourth algorithm uses color markers
and is employed as a reference system for the other three. All the al-
gorithms have been evaluated using two-dimensional video images of a
hand performing different finger movements on a flat surface. Results
about the accuracy, precision, latency and computer resource usage of
the different algorithms are provided. Applications of this research in-
clude human-computer interaction systems based on hand gesture, sign
language recognition, hand posture recognition, and gestural control of
music.

1 Introduction

The advances in technology and the widespread usage of computers in almost
every field of human activity are necessitating new interaction methods between
humans and machines. The traditional keyboard and mouse combination has
proved its usefulness but also, and in a more extensive way, its weakness and
limitations. In order to interact in an efficient and expressive way with the com-
puter, humans need to be able to communicate with machines in a manner more
similar to human-human communication.

In fact, throughout their evolution, human beings have used their hands, alone
or with the support of other means and senses, to communicate with others, to
receive feedback from the environment, and to manipulate things. It therefore
seems important that technology makes it possible to interact with machines
using some of these traditional skills.

The human-computer interaction (HCI) community has invented various tools
to exploit humans' gestures, the first attempts resulting in mechanical devices.

Devices such as data gloves can prove especially interesting and useful in certain specific applications but have the disadvantage of often being onerous, complex to use, and somewhat obtrusive.

The use of computer vision can consequently be a possible alternative. Recent advances in computer vision techniques and availability of fast computing have made the real-time requirements for HCI feasible. Consequently, extensive research has been done in the field of computer vision to identify hand poses and static gestures, and also, more recently, to interpret the dynamic meaning of gestures [6][9]. Computer vision systems are less intrusive and impose lower constraints on the user since they use video cameras to capture movements and rely on software applications to perform the analysis.

In order to avoid the problem of complex and not reproducible high cost systems, this paper focuses on two-dimensional systems using a single simple video camera. Algorithms using projection signatures, the circular Hough transform, and geometric properties have been chosen and are compared to an algorithm using color markers. Color markers are used solely as a reference system to evaluate the accuracy and the precision of the other algorithms, the presence of markers being a non-desirable constraint on the user of such a system. All the algorithms have been implemented in EyesWeb using the Expressive Gesture Processing Library [1] together with newly developed blocks (available in EyesWeb 4). These algorithms are designed to track different joints of the hand and more particularly of the finger (finger intersections, fingertips). Knowledge about these points on a frame-by-frame basis can later be provided to other analysis algorithms that will use the information to identify hand poses (static) or hand gestures (dynamic). Finger tracking is therefore at the base of many HCI applications and it opens new possibilities for multimodal interfaces and gestural control of music.

The algorithms presented in this paper are inspired by the research on tabletop applications [7][8]. These kinds of applications are often limited to the use of one finger instead of using the information that can be provide by tracking all fingers. Furthermore, these applications often use specific and expensive hardware (infrared camera for example). In this paper we suggest alternative methods that can work with simple hardware, such as a low-cost webcam. We use methods that were traditionally used in static pose identification (e.g. contour, signature) to do dynamic tracking. The use of the Hough transform, on the other hand, was inspired by research in 3-dimensional tracking [4], but also by some of the previously mentioned tabletop applications. These applications use the specific geometric shape of the fingertip with various templates matching algorithms to locate fingers.

The first section of this article briefly describes and illustrates the EyesWeb implementation of the four algorithms. Next, the test procedures are explained. The third section presents the results obtained from each algorithm during the tests. Finally, the article concludes with a comparative discussion of the potential uses of the different algorithms.

2   Methods

All the algorithms were evaluated using two-dimensional images of a hand performing different finger movements on a flat surface. The videos were recorded by a single fixed camera with a frame rate of 25fps (frame per second), fixed gain and fixed shutter. The tests were run on a Pentium 4 3.06GHz with 1Gb of RAM under Windows XP operating system. In order to test the algorithms, the problems of finding the region of interest and of eliminating complex backgrounds were reduced by shooting only the hand region on a uniform dark background. The second line of figures 1, 2, and 3 illustrates the segmentation process. In this simplify case, it consists of converting the image to gray-scale, applying a threshold to segment the hand from the background (using the fact that the hand is light while the background is dark), and filtering with a median filter to reduce residual noise.

2.1   Projection Signatures

Projection signatures, are performed directly on the resulting threshold binary image of the hand. The core process of this algorithm is shown on line 3 of figure 1 and consists of adding the binary pixels row by row along a diagonal (the vertical in this case). Previous knowledge of the hand angle is therefore required. A low-pass filter is applied on the signature (row sums) in order to reduce low frequency variations that create many local maxima and cause the problem of multiple positives (more than one detection per fingertip). The five maxima thereby obtained correspond to the position of the five fingers.

2.2   Geometric Properties

The second algorithm is based on the geometric properties and, as shown on line 3 of figure 2, uses a contour image of the hand on which a reference point is set. This point can be determined either by finding the center of mass of the contour (barycenter or centroid) or by fixing a point on the wrist [11]. Euclidean distances from that point to every contour points are then computed, with the five resulting maxima assumed to correspond to the finger ends. The minima can be used to determine the intersections between fingers (finger valleys). The geometric algorithm also required filtering in order to reduce the problem of multiple positives.

2.3   Circular Hough Transform

The circular Hough transform is applied on the contour image of the hand but could as well be performed on an edge image with complex background if no elements of the image exhibit the circular shape of the fingertip radius. The circular Hough transform algorithm uses the fact that the finger ends and the
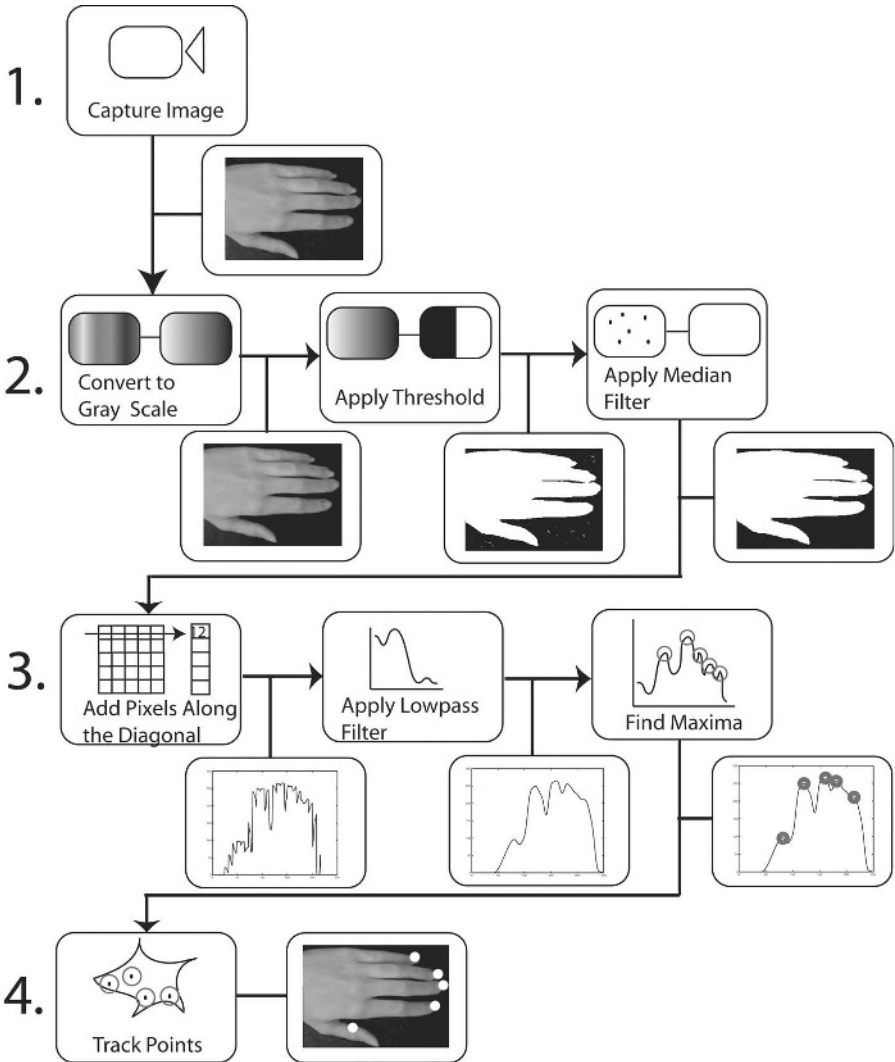
Fig. 1. Processing steps of the column signature algorithm

finger valleys have a quasi-circular shape while the rest of the hand is more lin-
early shaped. In this algorithm, circles of a given radius are traced on the edge
or contour image and regions with the highest match (many circles intersecting)
are assumed to correspond to finger ends and valleys (this process is illustrated
on line 3 of figure 3). Searched fingertips radius can be set manually or deter-
mined by an algorithm using the palm radius to fingertip radius proportion as
an estimate [2] [11] [4]. The circular Hough transform can find both finger ends
and valleys but, as opposed to the geometric algorithm, doesn't output them in
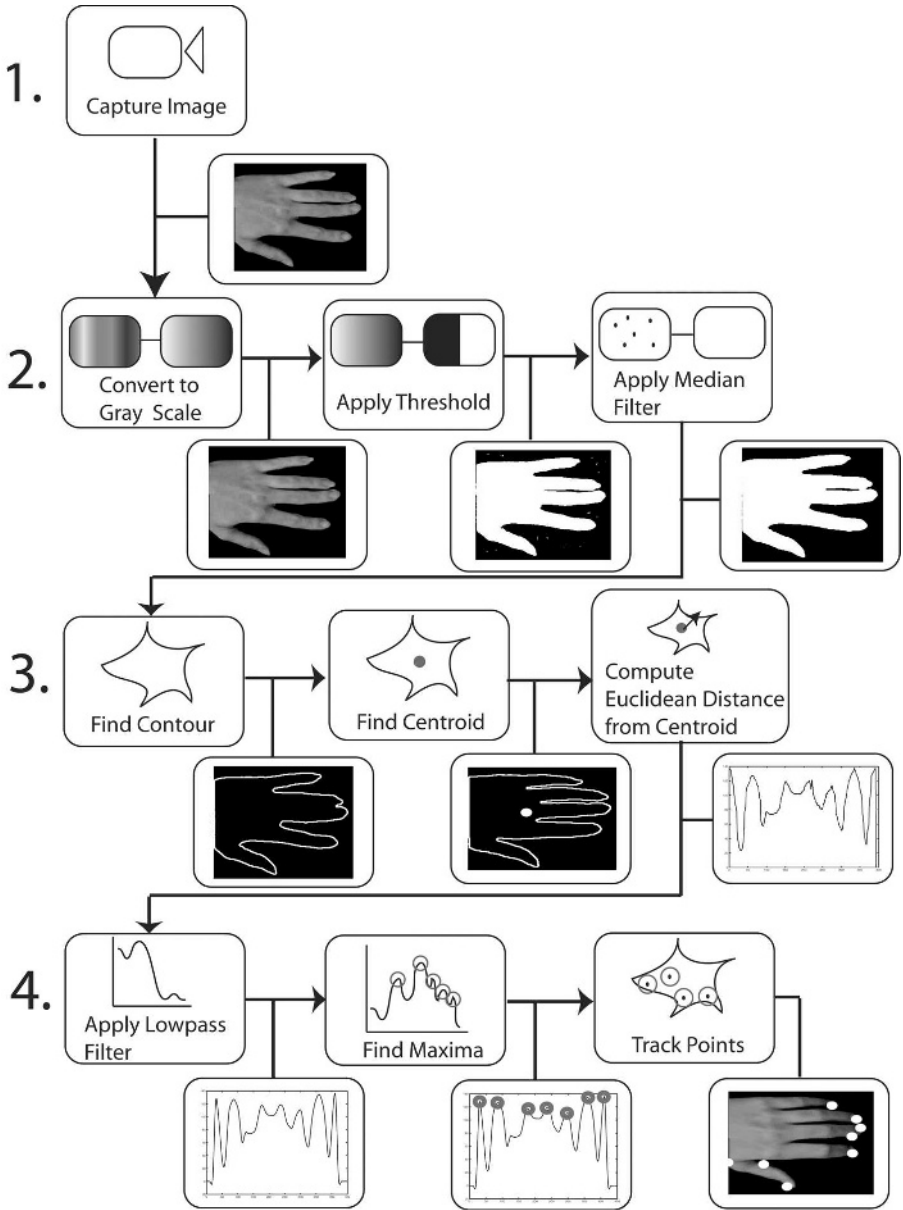
Fig. 2. Processing steps of geometric properties method

two distinct sets. Furthermore, the circular Hough transform requires filtering to eliminate false positives (detected regions that are not finger ends or valleys) that frequently appeared between fingers. As illustrated in line 4 of figure 3,
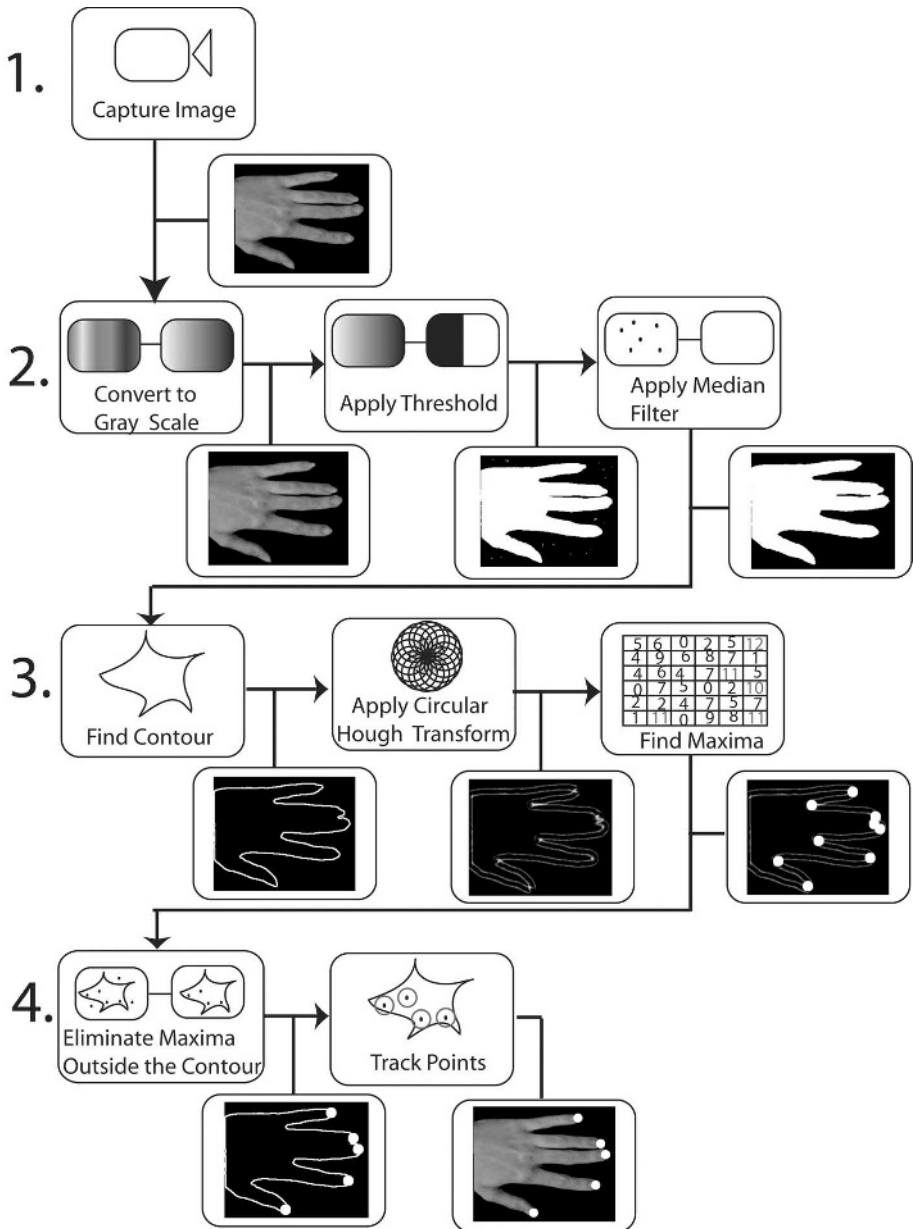
Fig. 3. Processing steps of circular Hough transform method

this can be done efficiently for finger ends by eliminating points that are found outside the contour image. The inconvenient is that the set of discard points contains a mix of finger valleys and false positive that cannot be sorted easily.

2.4   Color Markers

While the three previous algorithms rely only on the hand characteristics to find and track the fingers, the marker algorithm tracks color markers attached to the main joints of the fingers. Each color is tracked individually using color segmentation and filtering as illustrated in line 2 of figure 4. This permits the identification of the different hand segments. The marker colors should therefore be easy to track and should not affect the threshold, edge or contour image of the hand. Respecting these constraints makes it possible to apply all algorithms to the same video images and therefore to compare each algorithm degree of accuracy and precision with respect to the markers.
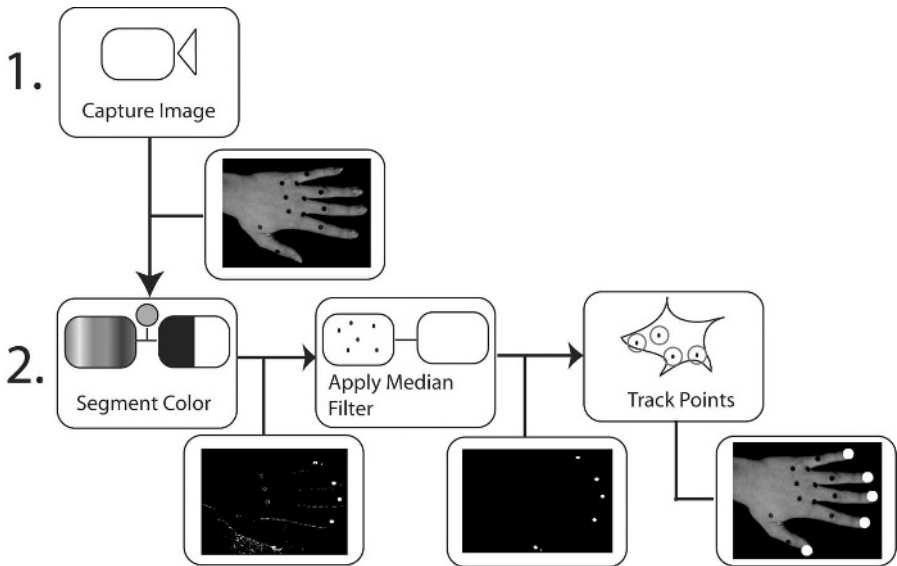
Fig. 4. Processing steps of color markers method

3   Tests

3.1   Accuracy and Precision

Accuracy and precision are important factors in the choice of a finger-tracking algorithm. The accuracy and precision of the different algorithms were determined with respect to the result obtained from the evaluation of the marker positions. To evaluate the accuracy and precision of the algorithms, the coordinates of 4 joints on each finger were tracked by applying the color tracking method (figure 4). Coordinates obtained with the three other algorithms were then related to the first set. The Euclidean distance between the marker and the closest point of each algorithm was computed. The accuracy of an algorithm

Fig. 5. Euclidean distance from the color marker for each of the three methods

can be determined by its distance from the marker. A curve close to zero denotes an accurate algorithm. The precision of an algorithm can be determined by observing the shape of the curve. A precise algorithm will exhibit an almost flat curve. Figure 5 presents the results obtained by tracking the tip of the small finger using each of the three algorithms. The values are compared to a marker placed at the center of the tip of the small finger. It can be observed that both the circular Hough transform and the geometric properties algorithm are precise algorithm since the distance between the marker and the point they return is almost constant. However, the circular Hough transform seems to be more accurate than the geometric properties. The average distance to the marker is really close to zero in the case of the circular Hough transform, but is approximately ten pixels in the case of the geometric properties.

The difference is mainly due to the fact that the geometric properties algorithm detects the extremity of the finger while the circular Hough transform finds the center and that where the markers are placed. In the case of the projection signatures, the detection of the fingers is robust but rough: the algorithm can only find the fingers and not a specific region of the finger like a tip or a valley. It can be observed in figure 5 that for an almost flat angle of the small finger, the accuracy is near twenty pixels (frame 0 and 160), for a small angle (between frame 50 and 160) it is approximately thirty pixels, and can go over a difference of sixty pixels for a large angle (after frame 160). This is due to the computation method, when the finger is angle, the end of the section that is in straight line with the palm will create a maximum and not the real finger end. This algorithm is consequently efficient only to find fingers or finger ends when the fingers are not angled. This algorithm is therefore neither accurate nor precise.

3.2   Latency and Resources Usage

The latency of each of the algorithms is determined by computing the delay between the evaluation of a frame and the output of its results. If the output rate is the same as the input rate (expressed in terms of the amount of time lapse between two input frames), no significant delay is generated by the evaluation part of the algorithm. In order to know the processing rate and the resource usage of the evaluation algorithm, all screen or file outputs were turned off. Table 1 displays the CPU (central processing unit) usage for each algorithm. The range is the observed minimum and maximum CPU usage percent throughout the duration of the test. The mode is the most frequently observed percentage. Table 1 shows that all the algorithms can be used in real time since no significant

Table 1. CPU usage of the three methods

Input Rate	Algorithms	CPU Usage Range	CPU Usage Mode	Output Rate
33 ms	Projection Signatures	10-18%	15%	33 ms
	Circular Hough Transform	38-77%	55%	
	Geometric Properties	16-45%	30%	

latency as been observed. Projection signature is extremely easy on computer resource with a mode of 15% of CPU usage and peaks ranging between 10 and 18%. Geometric properties is a bit more demanding with a mode of 30%. The poor performance of the circular Hough transform is probably due to the usage of the traditional algorithm [3] [10] that requires a lot of computation and storage for the accumulator cells, more modern implementations using probabilistic and heuristic approaches to optimize the algorithm performance exist [5] and are known to detect circles with the same degree of accuracy and precision.

4   Results and Discussion

We tested the previously presented algorithms with video recordings of the left and right hand of 5 users (3 females and 2 males, all adults). Results of these preliminary tests were coherent among all users and are qualitatively summarized in Table 2.

Table 2. Algorithms characteristics (+ → good to excellent, 0 → neutral to good, - → poor to neutral

	Projection Signatures	Geometric Properties	Circular Hough Transform	Color Markers
Locates fingers	+	+	+	+
Locates fingertips	-	0	0	+
Locates finger ends and valleys	-	+	+	+
Distinguishes between finger ends and valleys	-	+	0	+
Works with complex background	-	-	0	0
Works in real time (low latency)	+	+	+	+
Computer resources usage	+	+	-	+
Accuracy	-	+	+	+
Precision	-	+	+	+
Works with unknown hand orientation	-	+	+	+
Works with unknown fingertips radius	+	+	0	+

All the presented algorithms have succeeded, in various degrees, in detecting each finger. The projection signatures algorithm can only roughly identify a finger, but the circular Hough transform and geometric properties algorithms can find both finger intersections and finger end points, it is important to note that in the case where finger are folded, the end points dont correspond to the fingertips. The geometric properties algorithm outputs intersections and extremities in two distinct sets, but the circular Hough transform algorithm cannot make this distinction. The marker algorithm is the only one that can distinguish the various joints of the finger when different colors are used.

The projection signatures and geometric properties algorithms need a strong segmentation step prior to their application. The circular Hough transform, when combined with edge detection instead of contour, can work in complex environments, but some confusion can occur if other circular shapes of the size of the fingertip radius are present. Color markers can be used in complex backgrounds if the colors are properly chosen but are sensitive to light variation.

At 25fps all the algorithms output results without any significant delay; the input and output rate is the same. However, the circular Hough transform algorithm is much more demanding on CPU usage. This characteristic might limit its use when it is combined with pose and gesture recognition algorithms. The geometric properties and the circular Hough transform algorithms have similar and acceptable accuracy and precision values. The projection signatures algorithm cannot be used if these two characteristics are important.

The projection signatures algorithm can only be used in a controlled environment where the hand orientation is known and where finger angles dont vary to much from the straight line. The circular Hough transform algorithm needs previous knowledge of the fingertip radius or the palm radius. It can work in an environment where the distance from the video camera will change only if a method to estimate these radii is attached to it [2]. The geometric properties algorithm does not need any prior knowledge to be performed.

5   Conclusion

This article presented three algorithms to track fingers in two-dimensional video images. These algorithms have been compared to one another and evaluated with respect to a fourth algorithm that uses color markers to track the fingers. All the algorithms were implemented and tested in EyesWeb. Results relative to the precision, accuracy, latency and computer resource usage of each of the algorithms showed that geometric properties and circular Hough transform are the two algorithms with the more potential. The circular Hough transform should be preferred when a clean segmentation from the background is impossible while the geometric properties algorithm should be use when the fingertips radius is unknown and when information on both the finger ends and valley is required. Projection signature can be used as a fast algorithm to roughly obtained finger position. The choice of an algorithm should, therefore, depend on the application and on the setup environment. Future users should refer to the algorithms characteristics and constraints in table 2 to chose the appropriate one. It is also important to note that in this paper, the algorithms were tested alone and in a controlled environment. Consequently, the choice of an algorithm can also be influenced by the system in which it is supposed to work. As an example, the segmentation algorithm used in the pre-processing step and the pose or gesture algorithm used in the post-processing step can create constraints that will dictate the usage of a finger-tracking algorithm.

Acknowledgments

References

1. Camurri M., Mazzarino B., and Volpe G.: Analysis of Expressive Gesture: The EyesWeb Expressive Gesture processing Library, in Gesture-based Communication in Human-Computer Interaction. LNAI 2915, Springer Verlag (2004) 460-467
2. Chan, S. C.: Hand Gesture Recognition, [http://www.cim.mcgill.ca/~schan19/research/research.html], Center for Intelligent Machines, McGill University (2004).
3. Duda, S. R. D. and Hart, P. E.: Use of the Hough Transform to Detect Lines and Curves in Pictures in Communications of the Association of Computing Machinery, 15, 11-15, (1972).
4. Hemmi, K.: On the Detecting Method of Fingertip Positions Using the Circular Hough Transform in Proceeding of the 5th Asia-Pacific Conference on Control and Measurement, (2002).
5. Illingworth, J., and Kittler, J.: A Survey of the Hough Transform in Computer Vision, Graphics, and Image Processing, 44, 87-116, (1988).
6. Kohler M.:Vision Based Hand Gesture Recognition Systems, [http://ls7-www.cs.uni-dortmund.de/research/gesture/vbgr-table.html], Computer Graphics, University of Dortmund.
7. Koike, H., Sato, Y., and Kobayashi, Y.: Integrating Paper and Digital Information on EnhancedDesk: A Method for Realtime Finger Tracking on an Augmented Desk System, ACM Transaction on Computer-Human Interaction, vol. 8, no. 4, 307-322, (2001).
8. Letessier, J., and Brard F.: Visual Tracking of Bare Fingers for Interactive Surfaces, Seventeenth Annual ACM Symposium on User Interface Software and Technology, vol. 6, issue 2, 119-122, (2004).
9. Pavlovic V. I., Sharma R., and Huang T. S.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review, IEEE Transactions on pattern analysis and machine intelligence, vol. 19, no. 7, 677-695, (1997).
10. Schulze, M. A.: Cicular Hough Transform A Java Applet Demonstration, [http://www.markschulze.net/java/hough/], (2003).
11. Yörük E., Dutağaci H., and Sankur B.: Hand Biometrics, Electrical and Electronic Engineering Department, Boğaziçi University (to appear).

Captured Motion Data Processing for Real Time Synthesis of Sign Language

Alexis Heloir[1], Sylvie Gibet[1], Franck Multon[2], and Nicolas Courty[1]

[1] Université de Bretagne Sud, Laboratoire Valoria, Bâtiment Yves Coppens,
BP 573, 56017 Vannes Cedex, France
`firstname.surname@univ-ubs.fr`
[2] Université de Rennes 2, Laboratoire de Physiologie et de Biomécanique de
l'Exercice Musculaire, Av. Charles Tillon CS 24414, 35044 Rennes, France
`franck.multon@uhb.fr`

Abstract. This study proposes a roadmap for the creation and specification of a virtual humanoid capable of performing expressive gestures in real time. We present a gesture motion data acquisition protocol capable of handling the main articulators involved in human expressive gesture (whole body, fingers and face). The focus is then shifted to the postprocessing of captured data leading to a motion database complying with our motion specification language and capable of feeding data driven animation techniques.

Issues. Embodying a virtual humanoid with expressive gestures raises many problems such as computation-cost efficiency, realism and level of expressiveness, or high level specification of expressive gesture [1]. Here, we focus on the acquisition of motion capture data from the main articulators involved in communicative gesture (whole body, face mimics and finger motion). We then show how acquired data are postprocessed in order to build a database compatible with high level gesture specification and capable of feeding real time data-driven motion synthesis techniques. A recent automatic segmentation algorithm based on Principal Component Analysis (PCA) is then evaluated.

Motion acquisition protocol. The motion data acquisition protocol is designed to capture the whole range of articulators involved in order to produce human communicative gestures. This protocol relies on two complementary techniques, as shown in figure 1. The first technique aims at capturing facial and body motions and relies on a set of reflective markers placed on standardized anatomical landmarks and a network of 12 *Vicon-MX*[1] infrared cameras located all around the subject. The aim of the second technique is to capture finger motion thanks to pair of *Cybergloves*[2] measuring finger abduction and flexion. This technique is well-suited to finger motion, as it is robust by finger occlusions that may appear during the signing performance. The two sets of data acquired are post processed, synchronized and merged offline.

---

[1] `http://www.vicon.com/products/viconmx.html`
[2] `http://www.immersion.com/3d/products/cyberglove.php`
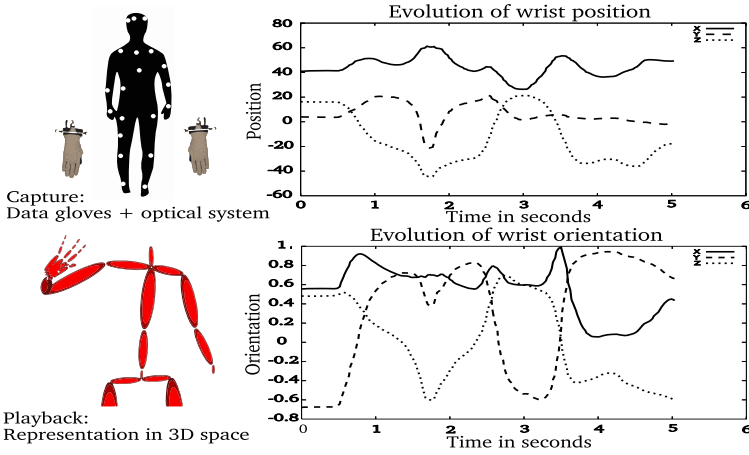
Fig. 1. Motion acquisition protocol

The resulting dataset describes the evolution along n frames of a skeleton hierarchy composed of k joints. For each joint i, a set of l_i degrees of freedom is defined, $1 \leq l_i \leq 3$. The size of a posture vector v can then be easily computed.

$$size(v) = \sum_{i=0}^{k} l_i$$

Table 1 sums up the composition of a posture vector according to our representation model.

Table 1. Detail of a posture vector

segment	coord type	number of joints	DOF per joint	size of segment subvector
body	angular	18	$1 \leq l \leq 3$	54
hand	angular	18	$1 \leq l \leq 3$	25
total	angular	36	—	79

Processing motion data. One of the early steps of data processing consists of segmenting motion into relevant chunks. Extracted chunks must be short enough to guarantee the synthesis of a wide range of new motions conveying sufficient meaning to comply with high level task oriented specification language [2].

Even though it has been shown that low level motion segmentation can be achieved in a straightforward manner [7][4], Hodgins and al. recently showed that higher level motion segmentation could be efficiently achieved thanks to the principal component analysis (PCA) approach. According to the results they presented [6], PCA segmentation method applied on simple motions representing typical human activities, such as walking, running, sitting, standing idle, etc.

achieved very good results: up to 80% precision call for simple body movements. This algorithm is based on the assumption that the intrinsic dimensionality of a motion sequence containing a single behavior should be smaller than the intrinsic dimensionality of a motion sequence containing multiple behaviors. Thus, from one motion sequence to another, the reconstruction error of the frames projected onto the optimal hyperplane of dimension r increases rapidly, for a fixed r. Figure 2 illustrates the motion transition detection between two hand configurations.
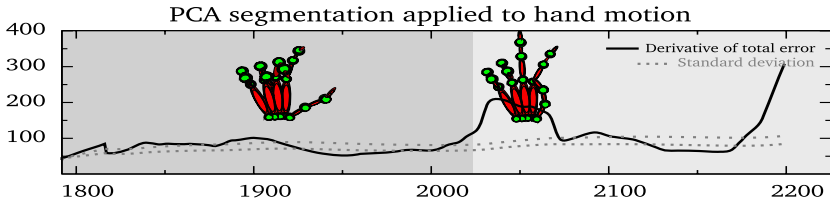
Fig. 2. Automatic segmentation using PCA. Cut is performed when derivative of error reaches three standard deviation from the average.

Evaluating the PCA approach to hand motion segmentation. We apply the PCA-based segmentation algorithm allocated to a sequence representing a non signer subject finger spelling French dactylologic alphabet [3]. The sequence is 7200 frames long with 120 frames per second. To carry out PCA, decomposition is thus performed on a 7200×25 matrix extracted from the total motion data and representing the right hand motion. According to our experiments, the ratio Er which indicates how much information is retained by projecting the frames onto the optimal r-dimensional hyperplane reaches acceptable range [6] when $r \leq 3$ for all the 20 segments we extracted manually from the alphabet spelling sequence. Further experiments led us to set the window parameter k originally fixed at 2 seconds to 1.3 seconds, considering that hand motion is much faster than body motion.
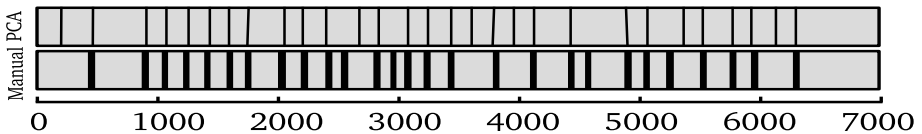
Fig. 3. Segmentation results

Results. In parallel to automated PCA based hand motion segmentation, a human observer manually segmented the finger spelling sequence by identifying probable zones of motion transition. Figure 3 compares how the two methods designated motion separation zones. The human observer identified 27 zones

while the PCA based motion segmentation algorithm identified 29 zones. Among those, 22 zones were overlapping.

Conclusion. We have presented a motion acquisition framework designed to manage several articulators involved in communicative gesture and in sign language performance. We then rely on the data provided by this framework to evaluate a recent automatic motion segmentation technique based on principal component analysis of hand motion. This method proves to be capable of solving high level segmentation required by our needs. In the near future, we wish to extend this technique to the whole of upper body motion. In parallel, we would like to provide a better evaluation framework based on data acquired and annotated by French sign language specialists. Such a framework will provide us with the grounds required to perform reliable motion analysis and performance comparisons.

References

1. Gibet S., Courty N., Multon F., Pelachaud C. and Gorce P.: HUGEX: virtual humanoids embodied with expressive communicative gestures. Proc. of Journées Bilan ROBEA (2005), 233–238, Montpelier, France.
2. Gibet S., Lebourque T., Marteau P. F.: High level Specification and Animation of Communicative Gestures. Journal of Visual Languages and Computing, **12** (2001), 657–687.
3. Moody B., La langue des signes - Tome 1 : Histoire et grammaire. International Visual Theatre (I.V.T.). Éditions Ellipses.
4. Kovar L., Gleicher M. and Pighin F.: Motion Graphs. Proc. of Int. Conf. on Computer graphics and Interactive Techniques (2002), 473–482, San Ãntonio, USA.
5. Keogh F., Palpanas T., Zordan V., Gunopulos D. and Cardle M.: Indexing Large Human-Motion Databases. Proc. of Int. Conf. on Very Large Data Bases (2004), 780–791, Toronto, Canada.
6. Barbič J., Safonova A.,Pan JY, Faloutsos C, Hodgins J.C. and Pollard N.S.: Segmenting Motion Capture Data into Distinct Behaviors. Proc. of Graphics Interface (2004), 185–194, London, Ontario, Canada.
7. A. Fod, M. J. Mataric, and O. C. Jenkins.: Automated derivation of primitives for movement classification. Autonomous Robots, **12** (2002), 39–54.

Estimating 3D Human Body Pose
from Stereo Image Sequences

Hee-Deok Yang[1], Sung-Kee Park[2], and Seong-Whan Lee[1,*]

[1] Center for Artificial Vision Research, Korea University,
Anam-dong, Seongbuk-ku, Seoul 136-713, Korea
`{hdyang, swlee}@image.korea.ac.kr`
[2] Intelligent Robotics Research Center, Korea Institute of Science and Technology,
P.O. Box 131, Cheongryang, Seoul 130-650, Korea
`skee@kist.re.kr`

Abstract. This paper presents a novel method for estimating 3D human body pose from stereo image sequences based on top-down learning. Human body pose is represented by a linear combination of prototypes of 2D depth images and their corresponding 3D body models in terms of the position of a predetermined set of joints. With a 2D depth image, we can estimate optimal coefficients for a linear combination of prototypes of the 2D depth images by solving least square minimization. The 3D body model of the input depth image is obtained by applying the estimated coefficients to the corresponding 3D body model of prototypes. In the learning stage, the proposed method is hierarchically constructed by classifying the training data into several clusters with a silhouette images and a depth images recursively. Also, in the estimating stage, the proposed method hierarchically estimates 3D human body pose with a silhouette image and a depth image. The experimental results show that our method can be efficient and effective for estimating 3D human body pose.

1 Introduction

Recognizing body gesture by estimating human body pose is one of the most difficult and commonly occurring problems in computer vision system. A number of researches have been developed for estimating and reconstructing 2D or 3D body pose [1, 2, 4, 5]. In this paper, we solve the problem of estimating 3D human body pose using a hierarchical learning method. The differences of our approach are: the 2D depth images and their corresponding 3D positions of body components are used to learn and the depth images are used to overcome ill-pose problem due to the similar silhouette images are generated by different human's body pose.

2 Gesture Representation

In order to estimate 3D human body pose from continuous depth images, we used a learning based approach. If we have sufficiently large amount of pairs of a depth and

---

its 3D body model as prototypes of human gesture, we can estimate an input 2D depth image by a linear combination of prototypes of 2D depth images. Then we can obtain its estimated 3D body model by applying the estimated coefficients to the corresponding 3D body model of prototypes as shown in Fig. 1. Our goal is to find an optimal parameter set α which best estimates the 3D human body pose from a given depth image. The proposed method is based on the statistical analysis of a number of prototypes of the 2D images are projected from 3D human model. The depth image is represented by a vector $d = (d'_1,...,d'_n)^T$, where n is the number of pixels in the image and d' is a value of a pixel in the depth image. The 3D body model is represented by a vector $p = ((x_1, y_1, z_1),...,(x_q, y_q, z_q))^T$, where x, y and z are the position of body joint in the 3D world. Eq. (1) explains training data.

$$D = (d_1,...,d_m), P = (p_1,...,p_m), S = (s_1,...,s_m) \tag{1}$$

where m is the number of prototypes and $s = (s'_1,...,s'_n)^T$ is a silhouette image.

A 2D depth image is represented by a linear combination of a number of prototypes of 2D depth images and its 3D body model is represented by estimated coefficients to the corresponding 3D body model of prototypes by such as:

$$\tilde{D} = \sum_{i=1}^{m} \alpha_i d_i, \quad \tilde{P} = \sum_{i=1}^{m} \alpha_i p_i \tag{2}$$
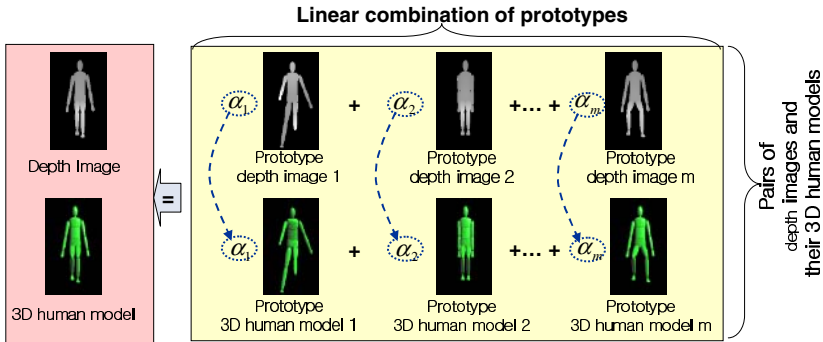
Fig. 1. Basic idea of the proposed method

3   Estimation of 3D Human Body Pose

To estimate 3D human body pose, we use three-level hierarchical model. In the first level, we estimate 3D human body with a silhouette history image(SHI) [6] applied spatio-temporal features which include continuous silhouette information. We compare the input silhouette image with the prototypes of 2D silhouette images, and select the prototype has the minimal distance. We use template matching to compare two silhouette images. After the first level, we estimate 3D human body with a silhouette image in the sub-cluster of current level. In the bottom level, we estimate 3D human body pose by using linear combination of prototypes of 2D depth images. Our estimation process consists of five-steps.

Step 1. Make a silhouette and a depth image applied spatio-temporal features from continuous silhouette images and normalize input data.
Step 2. Match a silhouette image and mean value of cluster at higher level of bottom level or estimate a parameter set to reconstruct silhouette image from the given depth image at bottom level.
Step 3. Estimate 3D human model with the parameter set estimated at Step 2.
Step 4. Compare the estimated 3D human model with the training data.
Step 5. Repeat Step 2, 3 and 4 for all levels of the hierarchical statistical model from top to bottom level.

4   Experimental Results and Analysis

For training the proposed method, we generated approximately 100,000 pairs of silhouette images and their 3D human models. The silhouette images are 170 x 190
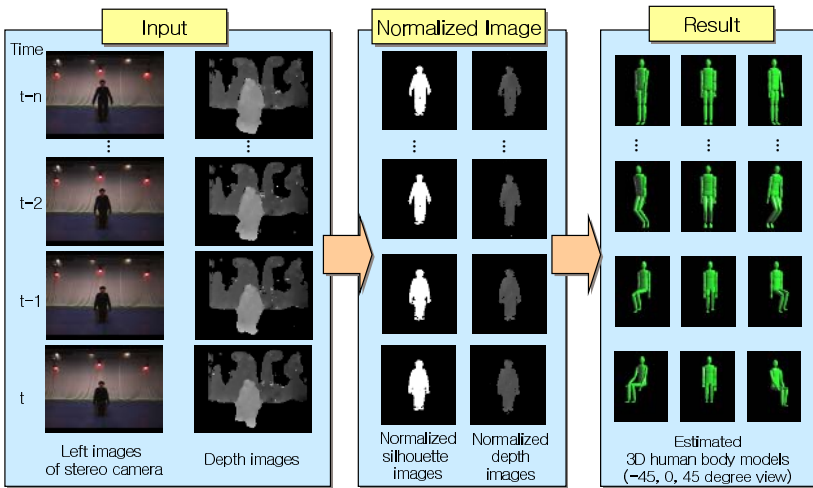
Fig. 2. Examples of the estimated 3D human body pose with sitting on a char sequence of the FBG database
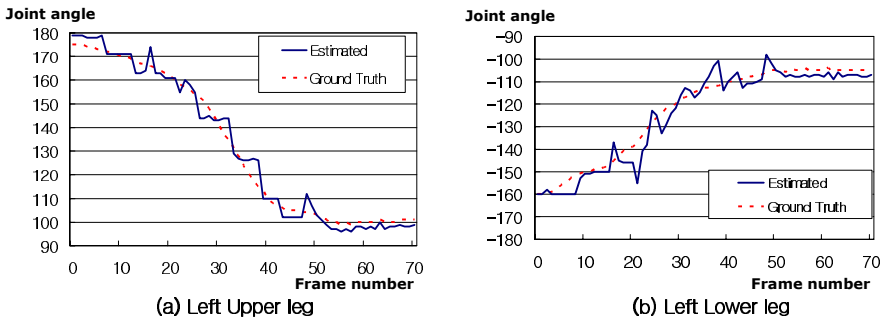
Fig. 3. Temporal curve of joint angles with the sequence in Fig. 2

pixels and their 3D human models are 17 joints in the 3D world. For testing the performance of our method, we used KU Gesture database [3]. Fig. 2 shows the estimated results which obtained in several images come from the FBG database at front view. The result of estimated 3D human body model represented front view, left 45 degree view and right 45 degree view of 3D human body model respectively. Fig. 3 shows the estimated angles of the left upper leg and the right upper leg with walking at a place sequence. As shown in Fig 3, the estimated joint angle at frame 7, 14, 27 are changed rapidly, because this region is the boundary of clustering algorithm.

5   Conclusion and Further Research

In this paper, we proposed an efficient method for estimating 3D human body pose from stereo image sequences using top-down learning. By applying depth information to estimate 3D human body pose, the similar pose in silhouette image cab be estimated different 3D human body pose.

Several interesting problem remains for further research. One topic is how to overcome the various size of real human body to a 3D human body model and the error of silhouette extraction. The other topic is how to solve extending the number of characteristic views. Using additive low-level information such as color, edge information and tracking extracted body component, we can analyze the relationship of human body components.

References

1. Agarwal, A., Triggs, B.: 3D Human Pose From Silhouette by Relevance Vector Regression. Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington D.C., USA (July 2004) 882-888
2. Bowden, R., Mitchell, T. A., Sarhadi, M.: Non-linear Statistical Models for 3D Reconstruction of Human Pose and Motion from Monocular Image Sequences. Image and Vision Computing, No. 18 (2000) 729-737
3. Hwang, B.-W., Kim, S., Lee, S.-W.: 2D and 3D Full-Body Gesture Database for Analyzing Daily Human Gestures. Lecture Notes in Computer Science, Vol. 3644, Hefei, China, (August 2005) 611-620.
4. Ong, E.J., Gong, S.: A Dynamic Human Model Using Hybrid 2D-3D Representations in Hierarchical PCA Space. Proc. of 10th British Machine Vision Conference, Nottingham, UK (Sep. 1999) 33-42
5. Rosales, R., Sclaroff, S.: Specialized Mapping and the Estimation of Human Body Pose from a Single Image. Proc. of IEEE Workshop on Human Motion, Texas, USA, (Dec. 2000) 19-24
6. Yang, H.-D., Park, S.-K., Lee, S.-W.: Reconstruction of 3D Human Body Pose Based on Top-Down Learning. Lecture Notes in Computer Science, Vol. 3644, Hefei, China, (August 2005) 601-610

Challenges in Exploiting Prioritized Inverse Kinematics for Motion Capture and Postural Control

Ronan Boulic[1], Manuel Peinado[2], and Benoît Le Callennec[1]

[1] VRLAB, Ecole Polytechnique Fédérale de Lausanne, EPFL,
1015 Lausanne, Switzerland
{Ronan.Boulic, Benoit.Lecallennec}@epfl.ch
http://vrlab.epfl.ch
[2] Escuela Politécnica, University of Alcalá, Spain
Manupg@aut.uah.es

Abstract. In this paper we explore the potential of Prioritized Inverse Kinematics for motion capture and postural control. We have two goals in mind: reducing the number of sensors to improve the usability of such systems, and allowing interactions with the environment such as manipulating objects or managing collisions on the fly. To do so, we enforce some general constraints such as balance or others that we can infer from the intended movement structure. On one hand we may loose part of the expressiveness of the original movement but this is the price to pay to ensure more precise interactions with the environment.

1   Introduction

Until now the exploitation of real-time motion capture of full body human movements has been limited to niche applications such as the expressive animation of a virtual character in a live show. Multiple factors hinder a wider adoption of full body movement as a powerful means for specifying and controlling the posture of a human mechanical model. Among others we can cite: the limited acquisition space and sensor precision, the spatial distortions, the high dimension of the posture space, and the modeling approximations in the mechanical model of the human body. These sources of errors accumulate and result in a low spatial control quality, hence making this approach not as usable as expected for evaluating complex human interaction with objects or with the environment. In the present paper we explore the potential of Prioritized Inverse Kinematics for motion capture and postural control, with Virtual Prototyping applications in mind. Our three immediate goals can be stated as follows: first reduce the number of sensors to improve the user comfort, second guarantee the correct recovery of the user manipulation of objects, and third integrate the postural control with automatic collision management. Our objective is to offer an intuitive and interactive control mode allowing any user to exploit body movements to quickly produce and manipulate human postures and movements. To do so, we enforce some general constraints such as balance or others that we can infer from the intended movement structure. In short, our approach is a trade-off that exchanges part of the expressiveness of the original movement in return for more precise interactions with the environment.

In the following sections, we first provide a classification of Inverse Kinematics techniques that have been used for real-time human motion capture, and we briefly recall the architecture of Prioritized Inverse Kinematics (PIK). Then we discuss the detection and handling of three problems that are critical for the success of intuitive user control with numeric IK: local postural singularities, self-collisions, and collisions with the environment. This latter case leads to an extension of the PIK architecture to minimize instabilities through the concept of observers. The paper ends with two case studies: recovering the movement of a musician from reduced sensor data, and automatically managing the (smooth) collisions with the environment in a reach task.

2   Background

2.1   IK Techniques for Motion Capture

Computer-assisted human motion capture can be traced back to the sixties as recalled by Sturman in his "Computer puppetry" article from 1998 [1]. The technical and computing complexity of this task has kept it mostly confined to niche applications such as the expressive animation of a virtual character in a live show. For these reasons the first systems were relying on a light exoskeleton mounted on the body and which delivered directly a measure of the main joint angles. Nowadays, the computing power being orders of magnitudes greater than in this pioneering time, new opportunities arise to exploit more sophisticated techniques for a greater comfort of the performer of the movement. Hence a broader range of applications may benefit from the integration of such type of interaction through body motion, e.g. for controlling a virtual mannequin in Ergonomics applications. We propose in this section a simple classification and comparison of the techniques used for reconstructing the skeleton configuration from sensor data. Modern sensors tend to be light and wireless and allow to obtain a 3D position (optical markers) sometimes together with a 3D orientation (magnetic sensors) with respect to a reference coordinate system. Evaluating the posture of a human body (i.e. its current joint state), from a set of such measures is often called the Inverse Kinematics problem (IK in short). There exist four major families of Inverse Kinematics algorithms:

Exact Analytic IK: Each body segment is equipped with a sufficient number of sensors so that its 3D position and orientation can be computed, hence leading to an unambiguous extraction of the joints' state from the relative transformation between successive segments [2]. Although fast, this approach still requires a significant number of sensors which reduces the comfort and increases its price.

Under Constrained Analytic IK: Only the 3D location of the pelvis, the torso and body extremities is captured (wrist, ankle, maybe the head). This allows to compute the 3D location of hip and shoulder joints and to obtain the arm and leg posture from an analytic formula described by Tolani [3]. In this approach, the swivel angle that exists around the line linking the shoulder to the wrist (and the hip to the ankle) cannot be determined without additional information; application-dependant heuristics are necessary to reconstruct a correct posture. Its fast computation makes this approach one of the most popular for motion retargeting.

Individualized Numeric IK: In the following approaches, the term *effector* denotes a coordinate system attached to the body (e.g. in one hand) that we want to attract to a goal position and/or orientation given by sensors. The Cyclic Coordinate Descent approach (CCD) searches for an optimal solution independently for each joint [4]. This algorithm has to be iterated to guarantee the convergence.

The Transpose Jacobian approach requires the repeated evaluation of the Jacobian J gathering the partial derivatives of the effector position and/or orientation with respect to the joint angle variables. In the case of an effector submitted to a *position constraint*, iteratively evaluating the product of the *position error* vector, noted Δx, with the *transposed Position Jacobian J^T* provides a joint variation vector that, once added to the current joint state, leads also to a convergence towards one solution posture [5].

Integrated Numeric IK: Here the Jacobian matrix J is viewed as a linearized approximation of the IK problem. We can obtain a locally optimal solution with its pseudo-inverse, noted J^+. For this approach to be valid only small error vectors Δx can be corrected at a time, therefore also requiring its integration within a convergence loop. Our tests have shown that despite its much higher computing cost per iteration, this approach is equivalent in terms of performance to the Transpose Jacobian approach [6] as it requires much less iterations to converge.

One key property of the pseudo-inverse is the possibility to build projection operators allowing to enforce a strict hierarchy among constraints. Indeed, all constraints do not have the same importance when searching for a solution in the joint space. For example, a position constraint on the centre of mass effector can ensure the balance of the posture [6], hence making it more critical than a position constraint attracting a hand effector towards a desired position. So it is important to be able to define priority levels among constraints, especially when they act in opposite directions in the joint space. In the next section, we examine how Prioritized Inverse Kinematics works and how to build the solution posture for a hierarchy of constraints.

2.2   Architecture of Prioritized Inverse Kinematics

Conceptual Analogy: To understand the following conceptual analogy it is necessary to be aware of the *redundancy* of the system, as the dimension N of the joint space is usually far greater than the dimension M of the constraint space. In practice it means that there generally exist an infinite number of joint variations that can achieve a given effector position variation to meet a constraint C. For example one can think of the set of postures that allow one hand to remain fixed in space.

In the drawings from Fig. 1 we illustrate the concept of priority hierarchy as it is conceptually enforced by Prioritized Inverse Kinematics. We represent the whole set of possible joint variations as a big ellipse while smaller shapes inside this ellipse represent joint variation subsets S_i that meet some constraints C_i. When two constraints are conflicting, their associated solution subsets do not intersect. In those cases the distance between the retained solution (a gray dot) and the closest point of any shape can be interpreted as a measure of the remaining error for the associated constraints. Fig. 1 illustrates the ideal case where the constraints do not conflict (left)
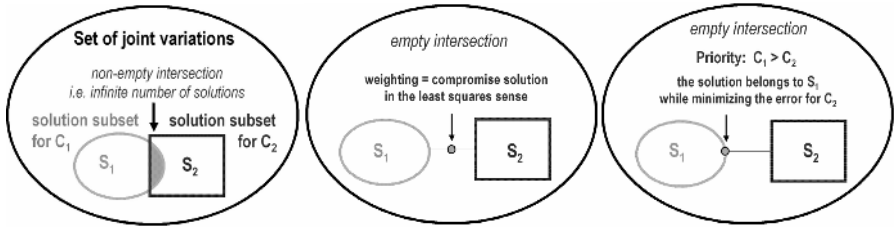
Fig. 1. The solution subsets S_1 and S_2 (respectively for constraints C_1 and C_2) may intersect (left); in case they don't the weighting approach provides a compromise solution (middle). The priority-based approach provides a solution belonging to the higher priority subset, here S_1, that minimizes the remaining error for C_2 (right).
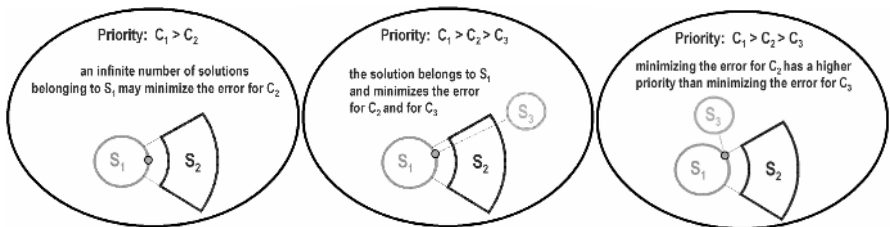
Fig. 2. An infinite number of solutions from S1 may equally minimize the error for the lower priority constraint C2 (left). If S3 is the solution subset of the lowest priority constraint, the solution belongs to S1 while minimizing the errors for C2 and C3 (middle). The C2 error minimization has a higher priority than the C3 error minimization (right).
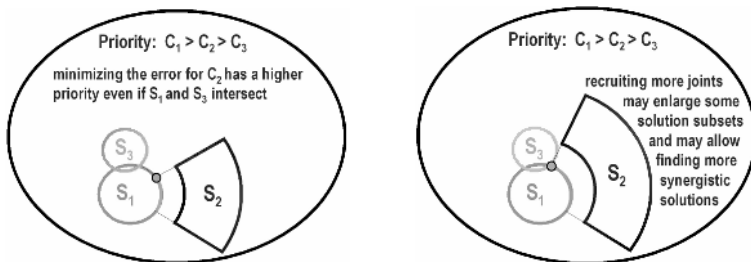
Fig. 3. The solution subset S_2 becomes larger when more joints are recruited for satisfying C_2; as a consequence a solution is found that also satisfies C_3 while respecting the hierarchy of constraints (left = small joint recruiting, right = large joint recruiting)

and compares the two possible approaches when they conflict: weighting minimization resulting in a compromise solution (middle) or prioritized solution favoring the constraint C_1 while minimizing the remaining error for C_2 (right).

In Fig. 2 we highlight the fact that an infinite number of solutions may exist within S_1 that minimize the error for C_2 (left). In case a third constraint C_3 of lowest priority becomes active and conflicts with C_1 and C_2, the retained solution minimizes both errors (middle). If the error minimizations also conflict, then the minimization of the

C_2 error has the higher priority than the one of C_3 (right). In the last conceptual drawings from Fig. 3 we want to stress that a solution subset S_i becomes "larger" as more joints are recruited for building the Jacobian associated to the constraint C_i. As a consequence, the retained solution may find solutions that meet more constraints, i.e. more synergistic solutions. For example in the left drawing a small recruiting for C_2 leads to a solution meeting only C_1 while a larger recruiting leads to a solution meeting both C_1 and C_3 (right).

Building the Prioritized IK Solution: We provide here only a brief overview as the full development of the equations can be found in [6]. Basically it relies on the equation derived by Hanafusa for two priority levels, and generalized by Slotine et al. for an arbitrary number of priority levels. Baerlocher has reduced the complexity of the projection operator update (all relevant references can be found in [6]).

$$[J_2(I - J_1^+ J_1)]^+ (\Delta x_2 - J_2(J_1^+ \Delta x_1))$$
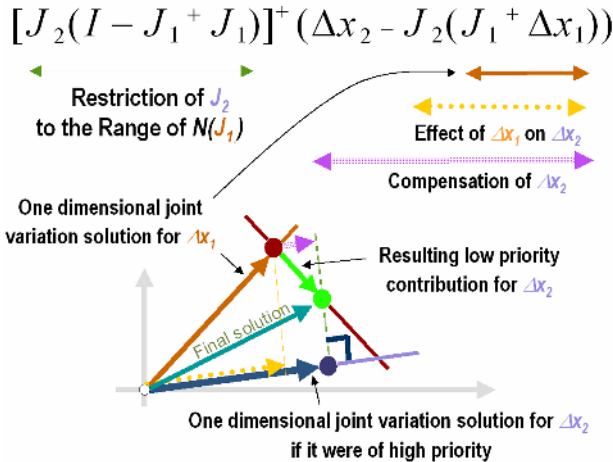
Fig. 4. Construction of the low priority contribution in a context of two scalar Cartesian constraints Δx_1 (high priority) and Δx_2 (low priority)

First Fig. 4 illustrates a simplified case of Hanafusa's solution with two one-dimensional Cartesian contraints C_1 (high priority) and C_2 (low priority), requesting respectively their constraint variations Δx_1 and Δx_2. It stresses how the one-dimensional high priority solution modifies the contribution of the low priority solution; first the term on the right of the equation is a compensation for what is already achieved in the high priority solution for the low priority constraint Δx_2. Then, it exploits the definition of the Null Space $N(J_1)$ of the linear transformation J_1 which states that any vector belonging to this subspace is transformed into the null vector by J_1. In concrete terms choosing any of these vectors as joint variation solution does not perturb Δx_1. So the purpose of the term of the left in Fig. 4 is to restrict the low priority Jacobian J_2 to the range of $N(J_1)$. Finally multiplying the right term with the pseudo-inverse of the left term provides the optimal low priority contribution that can be added to the high priority solution.

We now briefly describe how the two priority level architecture generalizes to an arbitrary number of priority levels p. For the sake of clarity we suppose that we have a set of p constraints $\{C_i\}$ each with an individual priority level indicated by i (from 1 to p, 1 being the priority of highest rank). Let Δx_i be the desired constraint variation for C_i. We denote by J_i the Jacobian matrix gathering the partial derivatives $\{\delta x_i/\delta\theta\}$. We build the joint space solution by accumulating the contribution of each priority level into to a joint variation vector $\Delta\theta$ as documented on Fig. 5. For each passage in this priority loop, we first compensate the desired constraint variation Δx_i by subtracting the influence of all higher priority tasks (step a).
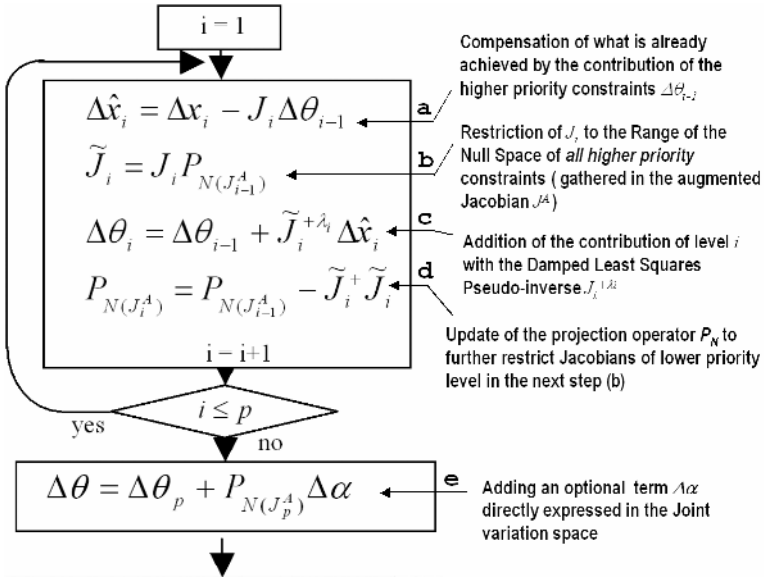
Fig. 5. Iterative accumulation of each priority level i contribution to the complete solution $\Delta\theta$, from the highest priority ($i=1$) to the lowest ($i=p$), with an additional lowest level contribution $\Delta\alpha$ directly expressed in the joint variation space (step e)

A second preparatory stage (step b) is to restrict the Jacobian J_i to the range of the Null Space $N(J^A)$ of all higher priority constraints (from 1 to i-1). This Null Space is defined through an additional Jacobian matrix, called the *Augmented* Jacobian and noted J^A that simply piles up all the individual constraint Jacobians from level 1 to level i-1 into one matrix. The associated projection operator $P_{N(J^A)}$ projects any vector from the joint variation space onto $N(J^A)$. Step c accumulates the contribution of the level i by multiplying the compensated constraint variation with the damped pseudo inverse of the restricted Jacobian. The damping factor λ_i is required for stability reasons around singular configurations (see [6] for details). The last step within the loop is the update of the projection operator $P_{N(J^A)}$ for restricting the next lower priority level constraint (step d). Once the priority loop is completed it is still possible to add the contribution of a joint variation vector $\Delta\alpha$ by projecting it with the lowest level projection operator (step e).

The priority loop is integrated within two other loops as shown on Fig. 6. Once the solution is available we check whether it violates one or more joint limits: If it is the case the detected joints are clamped and a new solution is search (see [6] for details). The external convergence loop is required as the desired constraint variations have to be bounded to remain within the validity domain of the first order approximation we make.
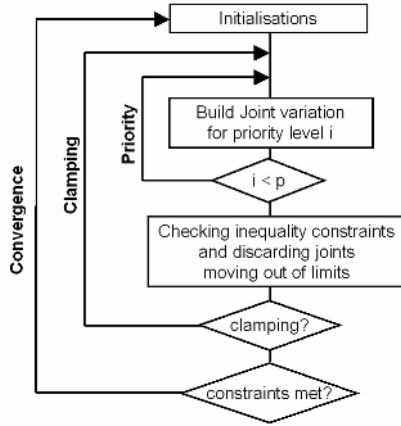
Fig. 6. Integration of the clamping loop between the internal priority loop (details in Fig. 5) and the external convergence loop

3   Three Critical Issues to Solve

Our experience has highlighted three key issues to solve for making Prioritized Inverse Kinematics more user-friendly: 1) the first order approximation greatly reduces the   ability to flex the arm and/or the leg when they are fully extended, 2) self-collisions and 3) the collisions with the environment. The treatment of these three problems should be automated as much as possible to relieve the end-user from the additional cognitive load of avoiding them. It is also an unpleasant experience to obtain counter-intuitive transient behaviors.

3.1   Local Postural Singularities

A singularity occurs when the rank of a Jacobian decreases due to the co-linearity of its column vectors. For example, in Fig. 7, when all segments of a chain align, their range space reduces from 2D to 1D. For a local postural singularity the rank of the Jacobian may not decrease; it is sufficient for two successive segments to be aligned to enter in such "sticky" postural state. This is easy to understand on Fig.7 (right) as no joint variation can be found to move away from the base (that's the intuitive way we understand a singularity) but also to move toward the base (this is rather counter-intuitive). As a consequence, once such a state is reached it tends to last during subsequent interactions, to the irritation of the user.
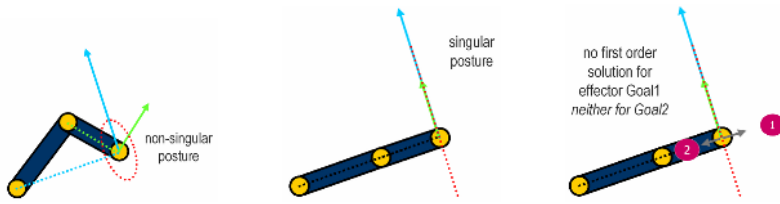
Fig. 7. The position of the chain tip effector can be controlled in 2D when the posture is non-singular (left); when in a singular posture its Jacobian becomes of rank 1 only (middle) thus preventing the chain tip to reach goals such as the points 1 or 2 (right)

Posture Attraction: The first solution that comes to mind is to attract the posture toward a slightly folded posture and enforce this attraction within the vector $\Delta\alpha$ at the lowest priority level (Fig 5e). However, being at such a low priority level, there is a risk that it doesn't get to be enforced. We illustrate this approach in the musician case study for handling the leg full extension.

The Concept of Observer: We propose instead the concept of *observer* to detect a local singularity and avoid it *when necessary*. Indeed a fully extended posture, even if singular, may serve some functional purpose and should not be always avoided. This is the reason why an observer can be viewed as a dormant effector that remains inactive as long as a necessary condition is not met. The triggering condition is to identify the case of the goal 2 in Fig. 7 (right), which consists in detecting that the segments are aligned and that there is a desired constraint variation oriented toward the base of the aligned segments. When such condition is met, a high-priority folding constraint is dynamically inserted in the hierarchy of constraints at the highest level.

3.2 Self-collision Avoidance

Even if individual joint limits are carefully designed to avoid unrealistic or unfeasible configurations, the fact is that some self collisions may occur when controlling a human body with a small number of sensors (the same problem arises with the under-constrained analytic IK). For example, in the musician case study the player's arms and torso often interpenetrate each other. Two approaches can be proposed:

Posture Attraction: Exploiting the low priority optimization task $\Delta\alpha$ can be a solution here too. All we have to do is adjust the goal posture so that the arms remain slightly away from the torso, thus preventing potential collisions from taking place. Note that this solution to the self collision problem does not penalize the performance, as we already exploit it to avoid a local singularity. On the negative side introducing such a goal posture may alter the expressiveness of the captured movement by imposing a preferred posture.

Repulsive Observers: Some dormant effectors can be created on the locations most likely to enter in collision with another body part, e.g. on the elbows. These constraints can be dynamically activated with a high priority when the effector location is penetrating other body parts, hence enforcing a strict Cartesian inequality constraint. Such effector is more effective when controlling only one translation dimension along the normal at the point of collision. This approach does indeed work,

but is significantly more expensive than the first one. We develop further the concept of observer in the next section.

3.3   Collision Avoidance with the Environment

Our key requirement is to provide a real-time method, therefore we have retained an approach capable of preventing the collision of a reasonable number of englobing solid primitives (sphere, segment). Instead of enforcing strict Cartesian inequality as outlined in the previous section, we prefer to extend the concept of observer and model collision management through *progressive* Cartesian inequality constraints. Basically, we want to anticipate the collision and alter the hierarchy of prioritized constraints before any hard collision occurs. For this purpose, any obstacle primitive is surrounded by a smooth collision zone in which we progressively alter the desired displacement of penetrating effectors and observers. A viscosity factor damps the displacement component along the normal of the obstacle but only when moving toward the obstacle. Fig 8. summarizes the integration of the observers management in the general Prioritized Inverse Kinematics architecture. For each new convergence step, we first process the regular effectors, then we check whether one or more observer enter the obstacles or their smooth collision zones. If it is the case they are temporarily upgraded to high priority effector with the altered displacement as desired constraint variation, and the current step is re-evaluated. Such re-evaluation is made until no *new* effector or observer is detected as colliding. Full details of this approach can be found in [7].
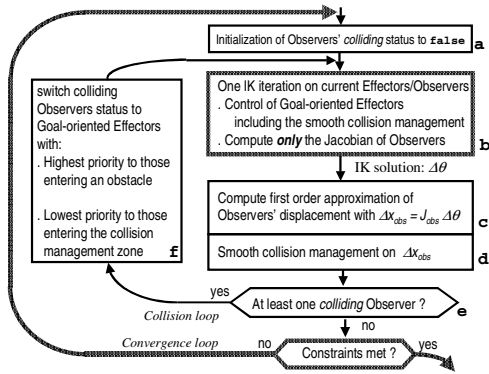
Fig. 8. Integration of the progressive Cartesian inequality constraints through observers in the general Prioritized Inverse Kinematics Architecture; both the priority and the clamping loops are included in stage b

4   Case Studies

4.1   Believable Musician Movement Recovery from Reduced Sensor Data

We faced the following challenge: a professional performer played the clarinet but her motion was captured using only six positional sensors: two for the ends of the

clarinet, and the remaining four for the head, shoulder, hip and knee of the player. All the sensors are located in the right side of the body. The amount of data provided by this set of sensors is insufficient for a traditional motion recovery method.

We have explored the use of the Prioritized Inverse Kinematics to reconstruct the original motion, at least to the extent that its main features are preserved. Table 1 gathers all the constraints that have been defined together with their priority level. We have already justified some of them in section 3; for the others the reader can find all the details in [8]. We simply illustrate the performances of our approach on a very specific expressive movement shown on Fig. 9 (first row of images). The second row contains images of the reconstructed movement in a real-time context where only one convergence step is allowed per sampled sensor data. One can notice a small lag of the hands with respect to their goal on the instrument; the feet constraints are also loosely met (more visible on the animation). These artifacts disappear when allowing three convergence step per sensor data as can be seen on the third row of images.

Table 1. Constraints and associated priority rank for the musician movement recovery

Constraint	Dimension	Total dimension	Priority rank
Keep the feet fixed on the ground (2 effectors/foot at heel and toe locations)	3/effector	12	1
Project center of mass between the feet	2	2	2
Place both hands on the clarinet	3/hand	6	3
Follow head sensor (position only)	3	3	4
Follow hip sensor (position only)	3	3	4
Attract toward preferred self-collision avoidance and singularity-free posture	Full joint space	Full joint space	5
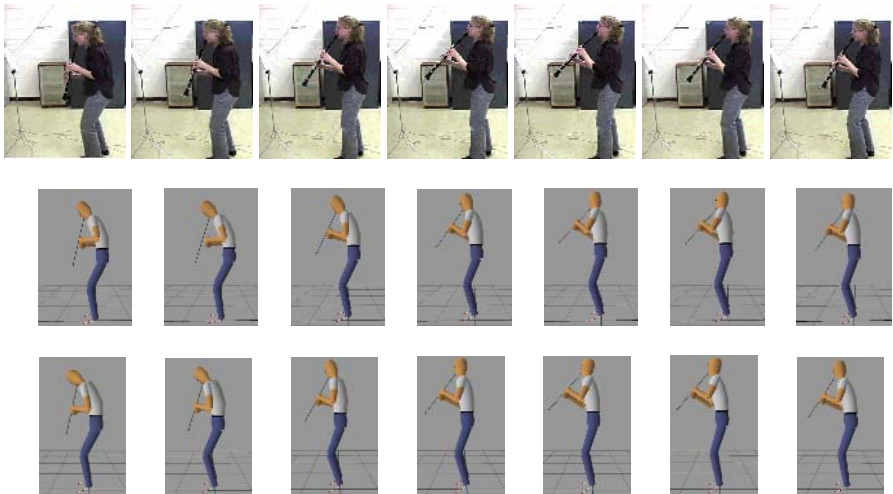
Fig. 9. An expressive movement (top), real-time (middle) and off-line (bottom) reconstruction

4.2 Progressive Cartesian Inequality Constraint

Fig. 10 shows a simple 15 joint chain reaching with its tip a goal located next to its base. A spherical obstacle (inner circle) is placed so close that a severe collision takes place if no collision avoidance strategy is applied (Fig. 10a). If we enforce only strict inequality constraints then this collision is prevented, as seen in Fig. 10b, but the chain gets arbitrarily close to the obstacle. A better result is achieved by the use of our progressive inequality constraint (Fig. 10c), in which the smooth collision zone (in-between the inner and the outer circle) strongly reduces the movement towards the obstacle while still allowing to reach the desired goal for the chain tip.
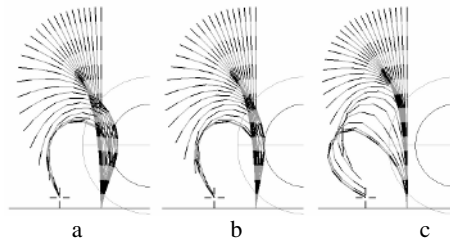
Fig. 10. (a) No obstacle avoidance. (b) Avoidance of hard collisions only. (c) Smooth inequality constraints added.

5 Conclusion

In this paper we have stressed three key issues faced by IK when used in interactive or real-time context with a reduced number of sensors. The musician study has confirmed the potential of our approach to recover believable movement from such a small set of sensor data if associated with prioritized constraints. Regarding the exploitation of the observer concept our preliminary results are very promising for the challenging context of virtual prototyping where collisions between the body of a virtual mannequin and a complex environment are bound to happen.

Acknowledgements

References

1. Sturman, D.: Computer Puppetry. IEEE CGA 18(1) (1998) 38-45
2. Molet, T., Boulic, R., Rezzonico, S., Thalmann, D.: An architecture for immersive evaluation of complex human tasks, IEEE TRA 15(3) (1999)

3. Tolani, D., Goswami, A., Badler, N.I.: Real-Time Inverse Kinematics Techniques for Anthropomorphic Limbs. Graphical Models 62(5) (2000) 353-388
4. Kulpa, R., Multon, F., Arnaldi, B.: Morphology-independent Representation of Motions for Interactive Human-like Animation. Computer Graphics Forum 24(3) (2005)
5. Welman, C.: Inverse Kinematics and geometric constraints for articulated figure manipulation, Ms Thesis, Simon Fraser University, 1993
6. Baerlocher, P., Boulic, R.: An Inverse Kinematic Architecture Enforcing an Arbitrary Number of Strict Priority Levels. The Visual Computer 20(6) (2004)
7. Peinado, M., Boulic, R.,Le Callennec, B., Meziat, D.: Progressive Cartesian Inequality Constraints for the Inverse Kinematic Control of Articulated Chains. In Proc. of Eurographics'05 Short Presentations, Dublin August-Sept. 2005
8. Peinado, M., Herbelin, B., Wanderley, M., Le Callennec, B., Boulic, R., Thalmann, D., Meziat, D.: Towards Configurable Motion Capture with Prioritized Inverse Kinematics. In SENSOR04, Proc. of the Third International Workshop on Virtual Rehabilitation (IVWR'04), (2004) 85-97, Lausanne

Implementing Expressive Gesture Synthesis for Embodied Conversational Agents

Björn Hartmann[1], Maurizio Mancini[2], and Catherine Pelachaud[2]

[1] Stanford University, Computer Science Department, Stanford CA 94305, USA
bjoern@cs.stanford.edu
[2] LINC-LIA, University of Paris-8, 93100 Montreuil, France
{m.mancini, c.pelachaud}@iut.univ-paris8.fr

Abstract. We aim at creating an expressive Embodied Conversational Agent (ECA) and address the problem of synthesizing expressive agent gestures. In our previous work, we have described the gesture selection process. In this paper, we present a computational model of gesture quality. Once a certain gesture has been chosen for execution, how can we modify it to carry a desired expressive content while retaining its original semantics? We characterize bodily expressivity with a small set of dimensions derived from a review of psychology literature. We provide a detailed description of the implementation of these dimensions in our animation system, including our gesture modeling language. We also demonstrate animations with different expressivity settings in our existing ECA system. Finally, we describe two user studies that evaluate the appropriateness of our implementation for each dimension of expressivity as well as the potential of combining these dimensions to create expressive gestures that reflect communicative intent.

1 Introduction

Embodied Conversational Agents (ECAs) are virtual embodied representations of humans that communicate multimodally with the user through voice, facial expression, gaze, gesture, and body movement. Effectiveness of an agent is dependent on her ability to suspend the user's disbelief during an interaction. To increase believability and life-likeness of an agent, she has to express emotion and exhibit personality in a consistent manner [1]. Human individuals differ not only in their reasoning, their set of beliefs, goals, and their emotional states, but also in their way of expressing such information through the execution of specific behaviors. During conversation, expressivity may manifest itself through gesture selection – *which* types of gestures are displayed – as well as through manner of execution – *how* they are displayed. In this paper we present an augmentation to our GRETA agent architecture that allows for parametric control of the qualitative aspects of gestures. Execution manner may depend on emotion, personality, culture, role and gender as well as on semantic and pragmatic aspects of the utterance itself in complex ways. We restrict our attention to generating phenomenologically accurate behaviors without claiming to correctly

represent internal processes. The paper is structured as follows: related work is reviewed in section 2, and our method for parametrizing gesture expressivity is reviewed in section 3. After outlining the GRETA architecture in section 4, we devote the majority of the paper to a description of the implementation of the expressivity parameters in section 5. We conclude by describing the results of two evaluation studies of our system and pointers to future work in sections 6 and 7.

2   Related Work

Research in gesture synthesis can be divided into systems that address the problem of gesture selection and systems that address gesture animation. Gesture selection for agents has mostly been concerned with semantic aspects of human gesturing, often following McNeill's method of classification [2]. Cassell et al. select suitable non-verbal behaviors to accompany user-supplied text based on a linguistic analysis [3]. Tepper et al. cross the boundary towards gesture animation by automatically generating iconic gestures from a parametric model [4]. Noot and Ruttkay address the need for inter-subject variability in GESTYLE [5], which chooses between atomic behaviors based on 'style dictionaries.'

Gesture animation is concerned with realistic movement generation of an agent's arms and hands from an abstract gesture representation language [6, 7]. Often, inverse kinematics techniques are used to calculate wrist trajectories [8]. Other systems allow for modification of existing body animations [9]. Of these, EMOTE by Chi et al. [10] is most closely related to our work as it also introduces an intermediate level of parametrization to obtain expressive gestures. EMOTE implements Laban principles from the dance community, while our system relies on psychology literature to obtain a set of expressivity parameters. EMOTE acts as a generic filter on pre-existing behaviors, while we tie behavior modification into the synthesis stage of gesturing.

3   Expressivity Parameters

We conducted a literature review of social psychology to arrive at a dimensional characterization of expressivity in human bodily movement. We regard an intermediate level of behavior parametrization as a useful enabling tool to facilitate the mapping of holistic, qualitative communicative functions such as mood, personality, and emotion to low-level animation parameters like joint angles. Our approach is driven by a perceptual standpoint – how expressivity is perceived by others. That is, we focus only on the surface realizations of movement and do not attempt to model underlying muscle activation patterns.

Based on an aggregation of the most pertinent studies [11, 12, 13] and our analysis of a gesture corpus [14], we propose to capture gesture expressivity with a set of six attributes which we describe below in qualitative terms. As part of an individualized agent's definition, personal default values for the expressivity attributes are defined.

- *Overall Activation*: quantity of movement during a conversational turn (e.g., passive/static or animated/engaged).
- *Spatial Extent*: amplitude of movements (amount of space taken up by body)
- *Temporal Extent*: duration of movements (e.g., quick versus sustained actions)
- *Fluidity*: smoothness and continuity of overall movement (e.g., smooth versus jerky)
- *Power*: dynamic properties of the movement (e.g., weak versus strong)
- *Repetition*: tendency to rhythmic repeats of specific movements.

Each of the attributes is float-valued and defined over the interval $[-1, 1]$, where the zero point corresponds to the actions our generic agent without expressivity control would perform. *Overall Activation* is float-valued and ranges from 0 to 1, where 0 corresponds to a complete absence of nonverbal behavior.
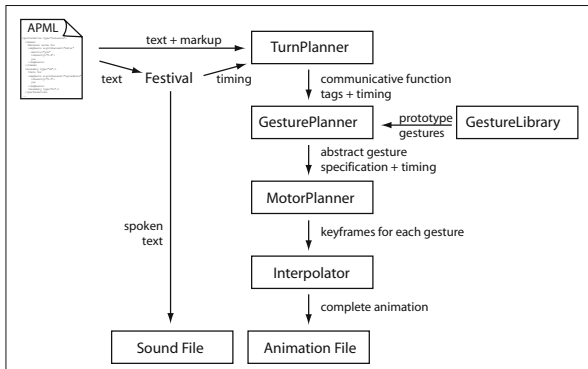
Fig. 1. Agent architecture outline

4   Expressive Agent Architecture

GRETA, our multimodal agent, interprets utterance text marked up in APML with communicative functions [15] to generate synchronized speech, face, gaze and gesture animations. The engines produce animation data in MPEG4-compliant FAP/BAP format, which in turn drive a facial and skeletal body model in OpenGL. We briefly review GRETA's *GestureEngine* [6] (see Fig. 1) here to clarify where expressivity modifications are performed. *GestureEngine* first performs text-to-speech conversion through Festival [16] which provides necessary phoneme timing for synchronizing gesture to speech. Communicative function tags which are candidates for gesture matching are extracted in the *TurnPlanner*. The *GesturePlanner* matches communicative function tags to a library of known prototype gestures and also schedules rest phases when arms are retracted to the body. The *MotorPlanner* then concretizes abstract gestures by calculating key frame joint angles and timing. Finally, a bank of different *Interpolators* generate in-between frames to complete the animation.

To enable the thus-far generic, deterministic architecture for expressivity control, we augmented different stages of the architecture, which we will describe in the next section.

5 Implementation: Mapping Expressivity into Gesture Animation Parameters

Given a particular type of action and a set of values in the expressivity space, how can we modify non-verbal behavior production to communicate the appropriate expressive content? We first need a suitable representation for gestures. We strive to preserve the semantic value of each gesture during the expressivity modifications. We hypothesize that effective strategies have to adjust behavior on multiple levels – from abstract planning (which type of gesture to use and whether to use a gesture at all), via gesture phase-level modifications (whether or not to repeat a stroke), down to adjusting velocity profiles of key pose transitions.

In the following, let the variables *oac*, *spc*, *tmp*, *flt*, *pwr* and *rep* stand for the *Overall Activation*, *Spatial Extent*, *Temporal Extent*, *Fluidity*, *Power* and *Repetition* parameter values we are trying to express.

5.1 Example

We introduce a sample dialog in transcript and APML-annotated form that will help clarify the expressivity computations we perform later on. The dialog was transcribed (and slightly edited) from an interview with author/journalist Helen Gurley Brown on the Open Mind television show[1]. We selected the following short utterance – words that coincided with gesture strokes are underlined:

"Whatever works for <u>you</u>, that's for <u>you</u>. But please don't tell me what works for <u>me</u>. Would you just please mind your <u>own business</u> and I'll mind <u>my business</u> and let's get on with the rest of our lives."

In the video, Hurley Brown performs a deictic reference to the interviewer (*you*), overlaid with a beat on the second *you*. A deictic gesture to herself with both hands accompanies the word <u>me</u>. After that, a metaphoric rejection is expressed by moving the right arm from shoulder-level downwards and out (*your business*). Finally, a round object in front of her torso is circumscribed to denote *[her] business*. We encoded this segment in APML, but for the sake of brevity only reproduce the beginning here in Figure 2. Text to be spoken by the agent is highlighted in blue.

5.2 Gesture Specification Language

In the past, we devised an abstract keyframe based scheme for gesture synthesis [6]. The gesture specification language is a sequence of key poses of the action,

---

[1] Publicly available through the Internet Archive: http://www.archive.org/

```
01: <performative type="announce">
02:   <rheme>
03:     Whatever works for
04:     <emphasis x-pitchaccent="Hstar" deictic="you" intensity="0.4">you</emphasis>
05:     <boundary type="LH"/>
06:     thats for
07:     <emphasis x-pitchaccent="LplusHstar" intensity="0.4">you</emphasis>
08:     <boundary type="LL"/>
09:   </rheme>
10: </performative>
```

Fig. 2. APML Dialog

each of which describes wrist location, palm orientation and hand shape. Sets of key poses are grouped into the gesture phases defined by McNeill [2]. Our specification language was augmented by attributes defining which features of a gesture carry its semantic meaning and are thus invariable, and which features can be modulated to add expressivity. Description of the temporal aspect of each gesture was made implicit. Where previously kinematics were fixed through the frame times of the key frames, timing is now calculated using motion functions. Let us consider the gesture matched to the deictic pointing towards the user (line 4 of our APML script). This gesture consists of a simple arm movement that halts on the upper torso and a hand configuration that points at the conversation partner. The hand is not immediately retracted, but remains in a post-stroke hold. Figure 3 shows our encoding of this gesture gesture. To conserve space, frames have been arranged horizontally and are to be read from left to right.

```
STARTFRAME                          STARTFRAME                          STARTFRAME
  FRAMETYPE stroke_start              FRAMETYPE stroke_end                FRAMETYPE post_stroke_hold
  ARM XC:fixed YUpperC ZMiddle        ARM XC:fixed YLowerC ZMiddle        ARM XC:fixed YLowerC ZMiddle
ENDFRAME                              HAND form_open thumb_default         HAND form_open thumb_default
                                      WRIST FBInwards PalmTowards          WRIST FBInwards PalmTowards
                                      ADDNOISE                           ENDFRAME
                                    ENDFRAME
```

Fig. 3. Sample gesture definition script

The postfix `:fixed`, highlighted in red, indicates that a particular element of the gesture must not be modified by expressivity calculations. In the deictic reference, the agent's hand points towards the user who is facing the agent through the screen. Thus the agent should point straight outwards and not besides herself. We thus constrain the lateral X coordinate of the arm goal position to be in the center sector of McNeill's gesture space [2]. While the duration of a hold depends on synchronization with adjacent gestures and speech, we explicitly encode the presence of a hold since this feature carries semantic weight. Note the absence of explicit timing information. The Gesture Engine calculates default durations. While we lose fine grain control compared to earlier explicit timing information, we gain parametric control over gesture phases as we will describe in section 5.3.

5.3 Expressivity Parameters

We now go through each of the identified dimensions of expressivity and explain how they are implemented. The stages of Figure 1 will be referenced to explain where gesture modification takes place.

Overall Activation. A filter is applied at the level of the *GesturePlanner*, which assigns gesture prototypes to input text mark up tags. Each input tag carries an intensity attribute that captures how important stressing the tag's content through nonverbal signals is – in line 4 of our APML example, the deictic gesture has an intensity of 0.4. Communicative functions tags for which this activation attribute does not surpass a given agent's overall activation threshold are not matched against the behavior database and no nonverbal behavior is generated. Thus, in our example, the deictic gesture will only be matched if the agent has an overall activation threshold ≥ 0.4. A similar principle of activity filtering was presented and implemented by Cassell et al. in [3].

Spatial Extent. The space in front of the agent that is used for gesturing is represented as a group of sectors following McNeill's diagram [2]. Wrist positions in our gesture language are defined in terms of these sectors. We expand or condense the size of each sector through asymmetric scaling of sector center coordinates. For meaningful scaling, we establish sector center coordinates $\vec{p}_i$ relative to the agent's solar plexus. Then the modified sector centers are given by:

$$\vec{p_i'} = \begin{bmatrix} I & \vec{spc} \\ 0 & 1 \end{bmatrix} \cdot \vec{p_i} \qquad \texttt{with}: \qquad \vec{spc} = \begin{pmatrix} 1.0 + spc \cdot spc_{agent_{horiz}} \\ 1.0 + spc \cdot spc_{agent_{vert}} \\ 1.0 + spc \cdot spc_{agent_{front}} \end{pmatrix}$$

$spc_{agent_{horiz}}$, $spc_{agent_{vert}}$, and $spc_{agent_{front}}$ are individual scaling factors in the horizontal, vertical and frontal directions that can define individualized patterns of space use. To find the location of articulation for a gesture, we first compute a point in the dynamically resized gesture quadrant that matches the gesture definition. We then calculate joint angles needed to reach that target with the IKAN inverse kinematics package [17]. Figure 4 shows a neutral key pose and modified wrist locations for contracted as well as expanded spatial extent. We note that this technique is conceptually similar to EMOTE's kinematic reach space. While inverse kinematics are computationally expensive, they provide the only way of addressing arm movement in terms of goal positions. In a complex articulated joint chain such as a human arm, controlling forward kinematics (i.e., joint angles) directly yields non-linear and unpredictable results. In our example deictic gesture, increasing spatial extent will move the Y and Z goal coordinates away from the agent, while the X coordinate remains unchanged because of the `:fixed` constraint in the gesture definition.

Adjusting the elbow *swivel angle* (Tolani [17]) also directly changes the space taken up by the agent – extended elbows enlarge the body's silhouette. We can control each arm's IK swivel angle θ for every key position:
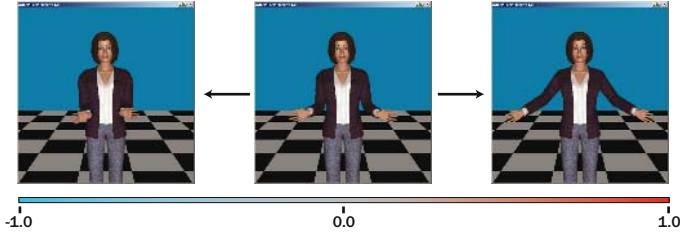
Fig. 4. Spatial Extent - the center image shows a neutral key pose. Arms are contracted for negative *spc* values (left) and extended for positive values (right).

$$\theta' = \{ \begin{array}{ll} \min(\theta \cdot (1.0 + 0.5 \cdot spc), \pi/2) & spc \geq 0 \\ \max(\theta \cdot (1.0 + 0.5 \cdot spc), 0) & spc < 0 \end{array}$$

Realistic default values for the swivel angle θ were established experimentally in various point of the reach space. These modifications are performed at the *MotorPlanner* stage.

Temporal Extent. Starting from the synchronicity constraint on the end of the gesture stroke to coincide with the stressed affiliate in speech [2], we can calculate preceding and proceeding frame times from invariant laws of human arm movement described in [8]. During the planning phase, the actual distance traveled by the wrist joint in space is approximated by linear segments through key points. The duration to complete each segment can be derived from a simplification of Fitt's law as

$$T = a + b \cdot \log_2(\|\vec{x_n} - \vec{x_{n+1}}\| + 1)$$

The value of the velocity coefficient b has been established as 10^{-1} for average speed movements by Kopp [7]. Using this value as a starting point, the speed of a gesture segment can be adjusted as follows:

$$b = (1 + 0.2 \cdot tmp) \cdot 10^{-1}$$

Since we can match keyframes to gesture phases, we can selectively amplify the stress of the gesture by increasing only the speed of the stroke to accentuate the gesture. Figure 5 shows arm position over time for a beat gesture for three different temporal extent parameter values.

Fluidity. This concept seeks to capture the smoothness of single gestures as well as the continuity between movements (the inter-gestural rest phases). We achieve low-level kinematic control through varying the continuity parameter of Kochanek-Bartels splines [18] used in the *Interpolator* component. Once again, this idea is close to EMOTE timing and fluidity control. In our implementation, we set the continuity parameter *cont* of the position interpolation spline for the wrist end-effector of each arm to equal the fluidity setting: $cont = flt$. The effect
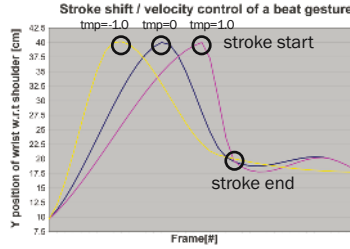
Fig. 5. Temporal Extent - plot of wrist position over time in one dimension. Keeping the timing of a gesture's stroke end fixed, the stroke start time is adjusted to control the speed of the stroke.
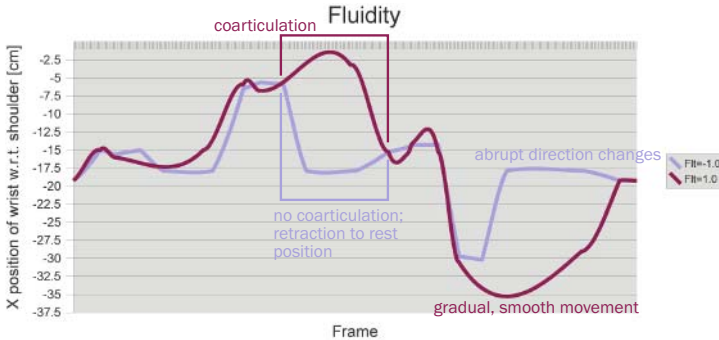
Fig. 6. Fluidity - this plot of wrist position over time in one dimensions shows different interpolation paths taken depending on the flt parameter

of this parameter on the shape of the interpolation spline is shown on the right side of Figure 6.

Fluidity also acts on the *GesturePlanner* level: larger fluidity increases the minimum timing threshold for retracting arms to a neutral position on the sides of the torso in between two gestures. During below-threshold pauses, arms are not retracted. Instead, two neighboring gestures are directly connected by interpolating between the retraction position of a previous gesture and the preparation position of the following gesture. In our example utterance, a low fluidity value would cause the agents arms to be retracted between the gestures accompanying the references to "*you*" and "*me*" (shown in transcript only, not in APML). A high fluidity setting would smoothly interpolate in the pause between gestures (as shown in the middle section of Figure 6).

Power. To visualize the amount of energy and tension invested into a movement, we again look at the dynamic properties of gestures. Powerful movements are expected to have higher acceleration and deceleration magnitudes. However, tense movements should exhibit less overshoot. This behavior is modelled with the tension and bias parameters of the position TCB-spline in the *Interpolator*:
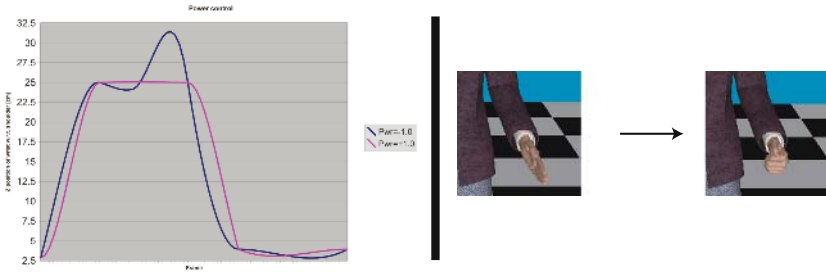
Fig. 7. Power - Wrist trajectory overshoot and hand shape are modified

$bias = pwr$ and $tension = pwr$. The bias parameter controls overshoot, while tension controls how sharply the spline bends at a keypoint. We also hypothesize that tense, powerful performances will be characterized by different hand shapes. If the configuration of the hand is not indicated as fixed in the gesture specification, high power settings will contract the hand towards a fist shape in the *GesturePlanner* stage. Figure 7 shows variation in wrist trajectories for minimum and maximum power settings on the left side, and an adjusted hand shape for a high power gesture on the right side.

Repetition. We have previously introduced the technique of stroke expansion [6] to capture coarticulation/superposition of beats onto other gestures. Stroke expansion repeats the meaning-carrying movement of a gesture so that successive stroke ends fall onto the stressed parts of speech following the original gesture affiliate. It is possible to control the extent of repetition by selectively increasing the 'horizon' or lookahead distance that the stroke repetition algorithm analyzes. In our example, the original speaker superimposed a beat onto the post-stroke hold of the deictic gesture for *you* during the second occurrence of the term. By increasing or decreasing the repetition parameter, we can encourage or discourage such superposition, respectively.

5.4   Aggregating Parameters

We now describe how combining expressivity parameters can modify gesture quality. Our system represents only a building block towards realizing affective action – exactly how motion quality is changed by the emotional state of an actor is still an open question in experimental psychology. Wallbott [12] described a partial mapping from emotional state to behavior quality, but much work remains to be done. Until a more comprehensive mapping is established, we use qualitative labels that are neutral with respect to emotion and personality, such as "abrupt." For abrupt action, "neutral" action is modified in the following ways: *Overall Activation* and *Spatial Extent* were disregarded (and thus left to the value 0) since abruptness is less apparent in the quantity of gestures or the amount of space taken up by those gestures. These two parameters are not

important to convey abruptness. *Temporal Extent* was increased to 1 to speed up the meaning carrying strokes of all gestures. *Fluidity* was decreased to -1 to create jerky, discontinuous velocity profiles of arm movements and to discourage coarticulation from one gesture to the next – the agent's arms are frequently retracted to a neutral position to create a disjoint performance. *Power* was set to a high value (1) to force a fist hand shape for beats and rapid acceleration and deceleration between gesture phases. Finally, *Repetition* was minimized (-1) since the rhythmic quality of a repeating movement counteracts the notion of abruptness. If we do not want to generate a strongly abrupt movement, we can generate *slightly abrupt* behavior by interpolating the pertinent parameters between "neutral" and "very abrupt" settings.

6 Evaluation

We conducted two evaluation tests. For the first test, we evaluated the following hypothesis: *The chosen implementation for mapping single dimensions of expressivity onto animation parameters can be recognized and correctly attributed by users.* 52 subjects were asked to identify a single dimension and direction of change in forced-choice comparisons between pairs of animation videos. 41.3% of participants were able to perceive changes in expressivity parameters and attribute those changes to the correct parameters in our dimensional model of expressivity. Recognition was best for the dimensions *Spatial Extent* (72.6% of modifications correctly attributed to this parameter) and *Temporal Extent* (73.8%). Modifications of *Fluidity* (33.9%) and *Power* (32.3%) were judged incorrectly more often, but the correct classification still had the highest number of responses. The parameter *Repetition* (28.0%) was frequently interpreted as *Power. Overall Activation*, or quantity of movement, was not well recognized. Overall, we take the results as indication that the mapping from dimensions of expressivity to gesture animation parameters is appropriate for the *Spatial Extent* and *Temporal Extent* dimensions while it needs refinement for the other parameters.

The second test with 54 subjects was conducted as a preference ranking task of four animations with different parameter combinations per trial to test the following hypothesis: *Combining parameters in such a way that they reflect a given communicative intent will result in more believable overall impression of the agent.* In each trial, one clip corresponded to the neutral, generic animation, two clips were variants of the chosen expressive intent (strongly and slightly expressive) and one clip had an inconsistent assignment of expressivity parameters. The subjects were asked to order the video clips from the most appropriate to the least appropriate with respect to the expressive intent. Participants in this second test preferred the coherent performance for the *abrupt* action described above over neutral and inconsistent actions as we had hoped. Similar results were obtained for the *vigorous* action. However, results were more ambiguous for our other test case - *sluggish* action. Two explanations are possible: the problematic implementation of some of the parameters may have led to unrealistic or

incoherent animation; alternatively, gesture modification alone may not be sufficient - it may have to be integrated with gesture selection to achieve truly believable expressive action. A person gesturing sluggishly might not use the same gesture types as a vigorously gesturing one.

7 Conclusion and Future Work

We have presented a computational model to add movement quality to communicative gestures. Six dimensions have been considered. We have evaluated the implementation of each of the six parameters individually and the ability of communicating a given intent with appropriate parameter combinations. We plan to refine our computational model, especially for the parameters that had low recognition rate. Control of dynamics is currently limited to modifying keyframe timing and interpolation quality. Physics-based simulation could provide a more suitable parametrization of movement, in exchange for higher computational cost. The conceptual interdependence of some dimensions, particularly Power and Temporal extent, also remains to be resolved. To ground further development in actual human performance, we will continue to work with annotated video corpora.

Acknowledgments

References

1. Loyall, A.B., Bates, J.: Personality-rich believable agents that use language. In Johnson, W.L., Hayes-Roth, B., eds.: Proceedings of the First International Conference on Autonomous Agents (Agents'97), Marina del Rey, CA, USA, ACM Press (1997) 106–113
2. McNeill, D.: Hand and Mind - What gestures reveal about thought. The University of Chicago Press, Chicago, IL (1992)
3. Cassell, J., Vilhjálmsson, H.H., Bickmore, T.: Beat: the behavior expression animation toolkit. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, ACM Press (2001) 477–486
4. Tepper, P., Kopp, S., Cassell, J.: Content in context: Generating language and iconic gesture without a gestionary. In: Proceedings of the Workshop on Balanced Perception and Action in ECAs at Automous Agents and Multiagent Systems (AAMAS). (2004)
5. Noot, H., Ruttkay, Z.: Gesture in style. In Camurri, A., Volpe, G., eds.: Gesture-Based Communication in Human-Computer Interaction - GW 2003. Number 2915 in LNAI. Springer (2004) 324

6. Hartmann, B., Mancini, M., Pelachaud, C.: Formational parameters and adaptive prototype instantiation for MPEG-4 compliant gesture synthesis. In: Proceedings of the Computer Animation 2002, IEEE Computer Society (2002) 111

7. Kopp, S., Wachsmuth, I.: Synthesizing multimodal utterances for conversational agents. The Journal of Computer Animation and Virtual Worlds **15** (2004)

8. Gibet, S., Kamp, J.F., Poirier, F.: Gesture analysis: Invariant laws in movement. In Camurri, A., Volpe, G., eds.: Gesture-Based Communication in Human-Computer Interaction - GW 2003. Number 2915 in LNAI. Springer (2004) 1–9

9. Neff, M., Fiume, E.: Modeling tension and relaxation for computer animation. In: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation, ACM Press (2002) 81–88

10. Chi, D., Costa, M., Zhao, L., Badler, N.: The EMOTE model for effort and shape. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co. (2000) 173–182

11. Wallbott, H.G., Scherer, K.R.: Cues and channels in emotion recognition. Journal of Personality and Social Psychology **51** (1986) 690–699

12. Wallbott, H.G.: Bodily expression of emotion. European Journal of Social Psychology **28** (1998) 879–896

13. Gallaher, P.E.: Individual differences in nonverbal behavior: Dimensions of style. Journal of Personality and Social Psychology **63** (1992) 133–145

14. Martell, C., Howard, P., Osborn, C., Britt, L., Myers, K.: FORM2 kinematic gesture corpus. Video recording and annotation (2003)

15. DeCarolis, B., Pelachaud, C., Poggi, I., Steedman, M.: APML, a mark-up language for believable behavior generation. In Prendinger, H., Ishizuka, M., eds.: Life-Like Characters. Cognitive Technologies. Springer (2004) –

16. Black, A., Taylor, P., Caley, R., Clark, R.: Festival. (http://www.cstr.ed.ac.uk/projects/festival/)

17. Tolani, D.: Inverse Kinematics Methods for Human Modeling and Simulation. PhD thesis, University of Pennsylvania (1998)

18. Kochanek, D.H.U., Bartels, R.H.: Interpolating splines with local tension, continuity, and bias control. In Christiansen, H., ed.: Computer Graphics (SIGGRAPH '84 Proceedings). Volume 18. (1984) 33–41

Dynamic Control of Captured Motions to Verify New Constraints

Carole Durocher, Franck Multon, and Richard Kulpa

Laboratoire de Physiologie et de Biomécanique de l'Exercice Musculaire,
University of Rennes 2, Av. Charles Tillon CS 24414,
35044 Rennes, France
{carole.durocher, franck.multon, richard.kulpa}@uhb.fr

Abstract. Simulating realistic human-like figures is still a challenging task when dynamics is involved. For example, making a virtual human jump to a given position requires to control the forces involved in take-off in order to reach a given velocity vector at the beginning of the aerial phase. Several problems are addressed in this paper in order to modify a captured motion while accounting from dynamics. The method exploits a point mass approximation of the body for the Inverse Dynamics stage during the contact phase and later to optimize new trajectories. First, accurate body segment masses are required to have access to external forces thanks to inverse dynamics. Second, those forces have to be adapted to make the resulting center of mass trajectory verify new constraints (such as reaching a given point at a given time). This paper also proposes a new formalism to encode force depending on time in contact phases (called impulse). Whereas classical biomechanical analyzes focus only on the peak of forces and on the contact phase duration, our formalism provides new data to characterize the shape of an impulse.

Keywords: take-off, optimization, dynamic control, constraints.

1 Introduction

Motion capture is widely used to guarantee realism in computer animation of virtual humans. Nevertheless, several problems still occur, ranging from signal processing to motion adaptation to new constraints. Given a set of external markers' trajectories, the main problem here is to propose a method to calculate the gestures that verify constraints imposed in a virtual environment: adapting the gestures according to the morphology, being at a given place at a given time, reaching a target with one or several part of the body, ensuring foot contact on the ground when the surface is not flat... The main applications deal with and are not limited to multimedia, film production, entertainment, virtual reality, training and education. Nevertheless, capturing a motion for each possible morphology and spacetime constraint is impossible so that specific techniques are required.

First approaches deal with minimizing a function that takes all the spacetime constraints into account [1][2][3]. Nevertheless, the required computation

time is very long due to the large number of degrees of freedom that are considered (up to 60 degrees of freedom). Hence, improvements focused on decreasing this complexity by using hierarchical approaches together with inverse kinematics [4][5]. An alternative consists in changing the representation used to code posture in order to deal with adimensional data in the Cartesian frame. With this proposal, the inverse problem is only summed-up to solving simple analytical inverse kinematics on body parts [6]. But, with all those techniques, only kinematics is considered and could result in unrealistic gestures, especially for highly dynamics motions (such as jumping, running fast…).

Some improvements were recently proposed to overcome the hierarchical approaches limitations by constraining the zero moment point in the feet-contact surface [7]. This technique makes it possible to adapt the gesture in order to respect balance. Another way to take balance into account is to control the position of the center of mass by using inverse kinetics [8]. The control of the center of mass is associated with high priorities whereas other constraints are associated to lower ones thanks to prioritized inverse kinematics [9][10]. Another advantage of such a technique is that it enables to make the center of mass follow a parabola in aerial motions. In order to go further this step in dynamically-sound motion adaptation [11] proposed to perform dynamic simulation on a simplified model with only few degrees of freedom and then to recalculate the remaining joints by inverse kinematics.

An alternative to all those gesture adaptation techniques is to directly use dynamical simulation on the complete skeleton [12]. In order to decrease the complexity of such a method, Principal Component Analysis is performed [13] and provides a restricted number of degrees of freedom to be controlled. The controller calculates torques by optimizing a cost function taking spacetime constraints into account while making the result resemble to captured motions. Scaling the gesture to a new character is also addressed [14]. Whatever the improvements, the computation time is not yet compatible with real-time animation in interactive and changing environments.

[15] proposed a force-based method to drive captured motions. First, the captured motion is segmented into sequences of stances and aerial phases, assuming that the center of mass is located at the root. Second, inverse dynamics provides forces at each time step. Third, an optimization process scales the first harmonic of those forces in order to verify new constraints. Finally, the posture is recalculated by optimizing the joints angles to make the center of mass follow the corresponding trajectory. The main advantage of such a method is that it separates the dynamics of the center of mass and the kinematics. This promising technique could be improved in order to decrease the computation time and to control more accurately the trajectory.

2   Overview

In this paper, we propose a new method to solve this problem (see figure 1). Given a captured motion, we compute the accurate center of mass (denoted COM) trajectory by first identifying the body segment masses of the original actor. Then, the method exploits a point mass approximation of the body for the
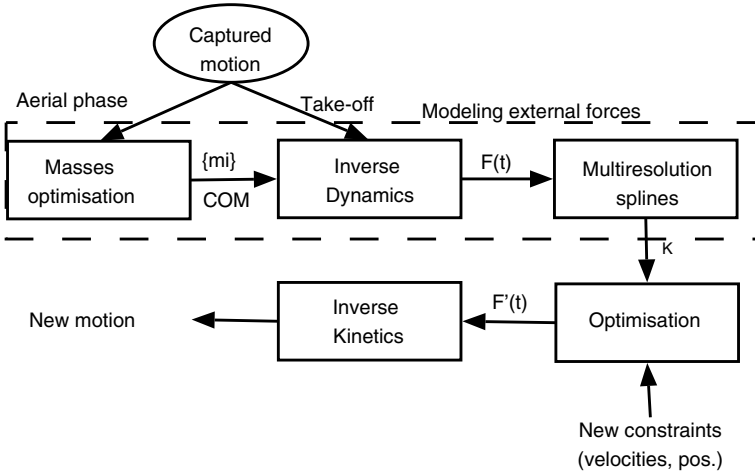
Fig. 1. Overview of the whole process developed in order to adapt the COM trajectory to new constraints

Inverse Dynamics stage during the contact phase and later to optimize new trajectories. Hence, inverse dynamics provides us with the external forces applied to the COM at any time. In biomechanics, take-off is defined as the integral of ground reaction forces during a contact phase. Take-off is widely studied in many physical activities given that it completely characterizes the velocity variations between the beginning and the end of take-off. Nevertheless, those biomechanical studies focus on the vertical peak force values [16] and take-off duration while this phenomenon is a continuous process over time. In our method, the shape of the resulting forces $F(t)$ is coded as multiresolution splines, as Liu et al. did for angular trajectories [3]. Multiresolution splines provide low-level control points, called keyforces (denoted K in the figure), and details. Thanks to this formalism, only a few parameters (the control points) are adapted rapidly to make the forces verify new constraints (such as jumping higher and taking the new character mass into account). The details are only time warped in order to follow the control points adaptations. As a consequence, we obtain new forces $F'(t)$ and a new COM trajectory that best verifies the constraints.

3   Modeling External Forces

The simplest mechanical model describing an articulated system consists in focusing on its COM. In this system, the external forces mainly involve the weight, the ground reaction force, contact forces and friction. The Newtonian laws enable to link the acceleration of the COM with those external forces at each time. For a given period of time, those laws become:

$$\int_0^L \left(\sum F_{ext} dt \right) = \int_0^L \frac{dp}{dt} dt = p_L - p_0 \qquad (1)$$

where F_{ext} are the external forces, 0 and L stand for the lower and upper boundaries of the considered time interval, p is the product of the mass m and the instantaneous velocity. As a consequence, this equation makes it possible to link the external forces to the variation of velocity between time 0 and L:

$$v_L = v_0 + \frac{\int_0^L \left(\sum F_{ext} dt \right)}{m} \tag{2}$$

where v_0 and v_L are respectively the COM velocity at time 0 and L. To retrieve the external forces applied during a captured motion, we have to process as follows. First, the total body COM G must be recovered. Second, acceleration of G is calculated for each frame. Then, Newtonian laws enable to calculate external forces as the mass multiplied by the acceleration of G.

Captured trajectories of external markers can be used to retrieve the body COM. To this end, joint centers should be retrieved in order to lower the skin sliding artifacts. We applied the method proposed in [17] to calculate the joint centers. According to joint centers, anthropometric tables [18] provide masses m_i, inertias I_i and local centers of mass location G_i for all the body segment. Several authors in biomechanics demonstrated that motion analysis (including external force evaluation) is very sensitive to the evaluation of such anthropometric data [19]. Hence, customization of such parameters is necessary to accurately calculate forces from motion capture. Some works in biomechanics [20] proposed to identify those parameters by minimizing the difference between forces calculated indirectly with motion capture and measurements provided by a forceplate. The motion is subdivided into two main phases: contact and aerial phase. During aerial phase, only the weight is acting on the COM assuming that friction is neglected. Hence, the external forces are well-known and Vaughan et al.'s approach [20] can be adapted:

$$f\left(\{m_i\}\right) = \prod_{i=1}^{N} \left(\frac{1}{p(m_i) + \epsilon} \right) \times \sum_{t=1}^{T} \left(\ddot{G}(\{m_i\}, t) - g \right)^2 \tag{3}$$

where $\{m_i\}$ is the set of body segment masses for all the segments S_i, G provides the global COM position according to time and a parameter set, g is gravity, $p(m_i)$ is the probability for a mass to be realistic (uniform distribution around the values proposed in anthropometric tables), ϵ is a small value to avoid division by 0, N is the number of body segments and T is the number of frames during the aerial phase. f is the cost function that should be minimized in order to identify the body segment masses. In this equation $N - 1$ masses are directly optimized whereas the last one is equal to the total body mass less the sum of the $N - 1$ other masses. This hypothesis enables to implicitly verify that the sum of all the body segment masses is equal to the total body mass. We performed specific experiments on 8 subjects to quantify the improvements brought by using such an optimization. The results demonstrate that the distance between the COM acceleration and gravity is decreased down to
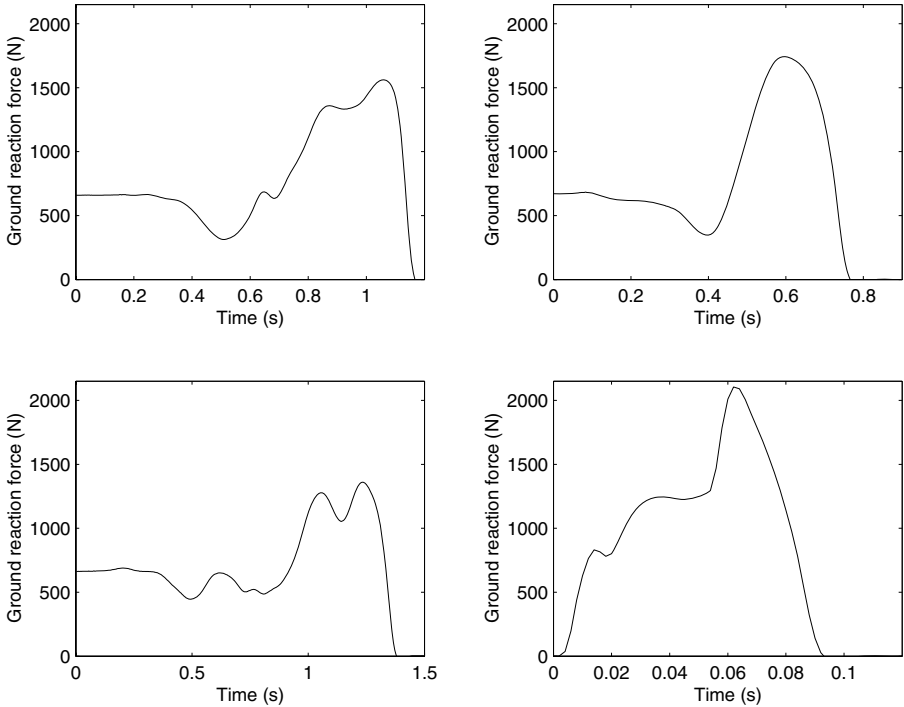
Fig. 2. Four examples of ground reaction forces in vertical jump (top-left), horizontal jump (bottom-left), kick with a jump (top-right) and volley-ball smash (bottom-right)

0.1 $m.s^{-2}$ (benefit ranging from 38% to 62%). As a result inverse dynamics applied in the contact phase provides forces similar to those measured by a force-plate (mean correlation equal to 0.95 and averaged root mean square error equal to 2.5 N).

Figure 2 depicts the vertical component of the external forces computed for several movements (vertical jump, high kicks, running, long jump) using this technique. Each component of the external force is a continuous function of time Fx(t), Fy(t) and Fz(t). Those functions can be modeled as multiresolution splines, as suggested by [3]. The multiresolution splines provide the global shape of the force together with additional details. The control points Φ_0 used for the global shape include the first and last value as well as all the points with null derivative. The details $\Psi_0(t)$ are coded as a discretized signal corresponding to the subtraction between the original signal and the one reconstructed with the control points. This process could be applied recursively by decomposing the details of level i into control points Φ_{i+1} and the remaining details $\Psi_{i+1}(t)$. In this work, we applied this process only on the first level to obtain a minimum set of parameters (only Φ_0 called keyforces and denoted K in the remainder of the paper, and $\Psi_0(t)$).

4   Optimization to Verify New Constraints

Contrary to motion adaptation techniques based on optimizing trajectories, we propose to optimize forces to solve a problem on a given period of time. The problem is to adapt a motion to new constraints:

 – An initial velocity v_0 that can be different from the one stored in the captured motion, depending on the previous actions that were performed,
 – A final velocity v_L that is imposed by the user in order to produce a desired aerial trajectory (such as jumping higher or to a larger distance),
 – And a set of keyforces that provide us the shape of the external forces between the beginning and the end of the sequence.

In that case, the problem could be formulated mathematically by solving equation 2. To this end, we can tune the keyforces K in order to solve the following problem:

$$m(v_L - v_0) = g(K) \tag{4}$$

where $g(K) = \int_0^L \left(\sum F_{ext}(K) dt \right)$, $F_{ext}(K)$ stands for the forces resulting from the use of the keyforces set K. This problem is non-linear but could be linearized locally, as it is currenlty performed in robotics to solve inverse kinematics problems [21]. Let v_0^0 and v_L^0 be respectively the initial and final COM velocities for the original motion (with keyforces K_0). The problem can be linearized in the neighborhood of K_0, as follows:

$$m(v_L - v_0) - m(v_L^0 - v_0^0) = J(K_0)\Delta K \tag{5}$$

with $\Delta K = K - K_0$ and $J(K_0)$ is the Jacobian of function g in K_0. Then, the solution is provided by inverting this equation:

$$\Delta K = J^+(K_0) \left(m(v_L - v_0) - m(v_L^0 - v_0^0) \right) \tag{6}$$

where J^+ is the pseudo-inverse of J. ΔK with this expression has a minimum norm. At this step, new constraints can be considered, such as:

 – minimizing the sum of square forces to avoid high forces values:

$$h_1 : min \left(\int_0^L F_{ext}^2(K_0 + \Delta K) dt \right) \tag{7}$$

 – and ensuring that the COM reaches the desired position at the end of the contact phase:

$$h_2 : min \left(\left(\int_0^t \left(\int_0^t \frac{1}{m} F_{ext}(\tau) d\tau + v_0 d\tau \right) + X_0 - X_L \right)^2 \right) \tag{8}$$

The solution of those constraints is searched in the null space of function g by using the projection operator $(I - J^+J)$. As a consequence, the solution is given by:

$$\Delta K = J^+(K_0)\left(m(v_L - v_0) - m(v_L^0 - v_0^0)\right) + (I - J^+(K_0)J(K_0))z \qquad (9)$$

where z is a function including constraints h_1 and h_2. As this computation is performed at each stance phase, the first keyforce at stance s is equal to the last keyforce at stance $s - 1$. If stance s is followed by an aerial phase, the last keyforce is also known and is equal to gravity with null derivative. Moreover, for all the intermediate keyforces the derivative is null due to our decomposition algorithm. As a consequence, only the time and the value for each intermediate keyforce has to be found in the system: from one to three control points in all the motions we experimented, which represents only 2 to 6 parameters for each axis. The Jacobian matrix that is computed numerically and the pseudo-inverse can be then calculated rapidly.

Once the force is defined for each time step, the trajectory of the COM can be computed. Let q_0 be the position of the COM at the beginning of the sequence. Giving q_0 and v_0, the velocity for each time step can be computed thanks to a two-times integration. Giving the COM trajectory and an initial motion, it could be possible to calculate new gestures by using inverse kinetics for each time step [9].

5   Results

In this section, we present the results obtained by optimizing the body segment masses and by modifying a COM trajectory in order to verify new initial and final velocities. First, let us consider the masses calculated by optimization according to external markers during the aerial phase. We performed our calculations on five subjects that jumped 10 times while equipped with 28 reflective markers. The markers' positions were stored by a Vicon370 motion capture system (Oxford Metrics) composed of 7 cameras cadenced at 60Hz. The subjects were asked to perform large and complex gestures during the aerial phase in order to evaluate our optimization method. We calculated the root mean square error between the COM acceleration (calculated with the optimized masses) and gravity. We also calculated this error without any optimization (with De Leva's anthropometric tables [18]) to evaluate the improvements provided by our method. The calculation of the COM acceleration during the aerial phase was improved with a factor ranging from 37% to 62%.

Second, we tried to estimate the quality of the optimization process used to modify the take-off. To this end, we selected a force curve obtained with a subject performing a vertical jump. Let K_0 be the keyforces identified in this force curve. Then, we selected another take-off (called goal-motion) performed by the same subject, with a different initial and final COM position and velocity. We optimized K_0 in order to verify the constraints of this goal-motion. The results are depicted in figure 3. In this figure, the original COM acceleration
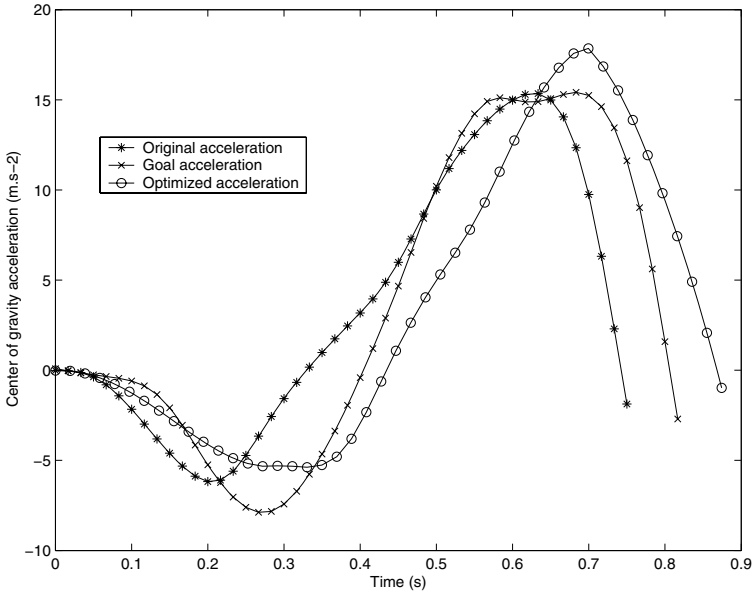
Fig. 3. Optimization (o) of a captured COM acceleration (*) in order to resemble to another captured one, called goal-motion (x)
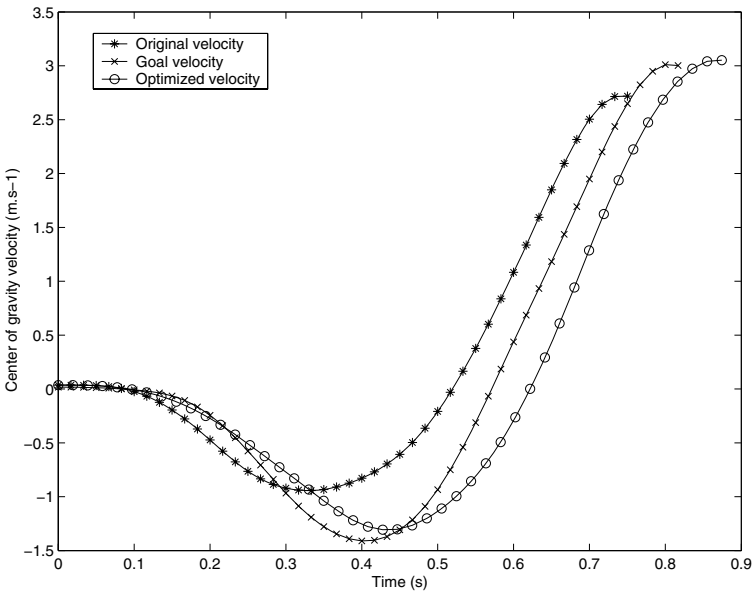
Fig. 4. Optimization (o) of a captured COM velocity (*) in order to resemble to another captured one (x)

is depicted with stars, the goal-motion is depicted with crosses and the result with 'o'. One can see that the COM acceleration of the goal-motion exhibits two maximum peaks in the final phase whereas the original one only exhibits one peak. Those two peaks in the goal-motion are linked to a time shift between the actions of the two legs. Nevertheless, with our method, it is impossible to make the optimized original take-off have two peaks given that only one keyforce is available in this neighborhood. As a consequence, in order to obtain the same integral as the goal-motion does, the system found a solution that involves a longer duration and a higher peak value.

Figure 4 describes the COM velocities for the original take-off, the goal-motion and the result provided by our method. One can see that the final velocity for the result is similar to the one observed for the goal-motion. Moreover, the shape of the result is similar to the one of the goal-motion, with a similar minimum value. As observed above, only the duration of the take-off is different.

As for velocities, the COM position for the result is similar to the one observed in the goal-motion (see figure 5).

As the differences between the two take-offs were low, we experimented our method with artificial goal-motions that involved higher differences. For example, we created artificially a goal-motion by multiplying the original take-off integral with a factor of 1.5 while preserving the original and final COM position. The COM velocity at the end of take-off was also multiplied by 1.5. Figure 5 provides the results obtained with such constraints. Intrinsically, our method preserves
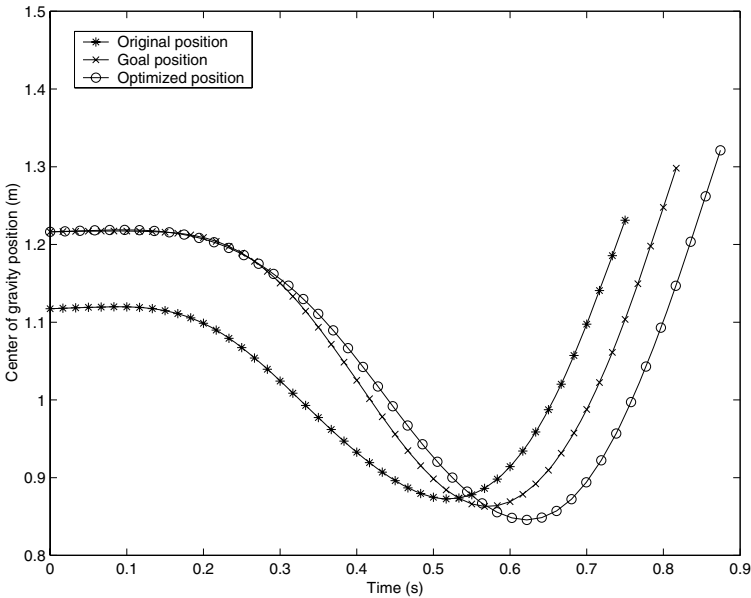
Fig. 5. Optimization (o) of a captured COM position (*) in order to resemble to another captured one (x)
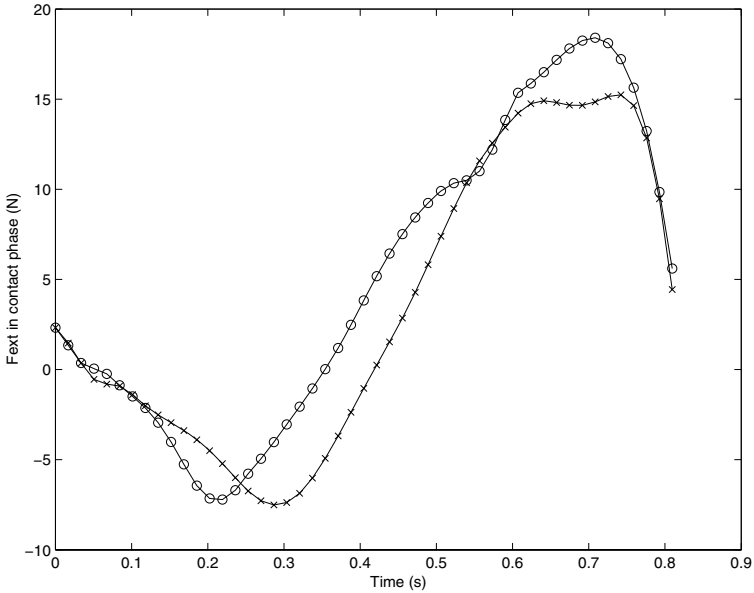
Fig. 6. Center of gravity acceleration in the original (x) and the optimized one (o) in order to multiply the final velocity by 1.5

the initial shape of the COM acceleration while adapting the total integral in order to reach a new velocity and position.

6   Discussion

In this paper, we addressed two main problems: automatic identification of body segment masses that enables to have access more accurately to forces than using anthropometric tables, and rapid adaptation of the external forces in order to verify new constraints. The main originality of this work is to model external forces acting on the COM as a set of keyforces (using a multiresolution spline decomposition). This choice enables to decrease the complexity of the optimization process. In Matlab (product of Mathworks), this process only requires around 0.4s on a Pentium 4 3GHz processor for a 1s take-off. This computation time could be significantly decreased if the algorithm would be coded in C^{++}. Future works will embed this method in an interactive animation engine in order to evaluate if it verifies real-time requirements.

Contrary to previous works [15], we do not optimize all the joint forces and torques to control the body segment movements because it requires too much computation time, including the use of a mechanical solver. Moreover, in this previous work, the parameters that were optimized were only the first harmonic of the external forces. In figure 2, one can see that those forces contain more than one main harmonic. Consequently, optimizing only the first harmonic should lead

to large approximations that might be unacceptable in some take-offs. In our method, we also use a simplified representation of those forces but splines seem more adapted to such non-periodic and complex signals. Moreover, although primary control points are optimized, the signal is completely reconstructed to evaluate the cost function to guarantee correctness.

Nevertheless, this technique has some limitations. The main limitation stems from instability if the beginning and the end of take-off are not accurately retrieved. Indeed, if we consider some samples after toe-off in the optimization process, the double integral of forces would not correspond to the actual COM position. The same way, if inaccurate initial position and velocity are used, the double integration would also engender unrealistic final position and velocity. To overcome partially those limitations, it would be necessary to accurately subdivide the motion into a contact and a flying phase. This automatic segmentation is not easy given that the COM acceleration rapidly falls to gravity at the end of take-off. An alternative should be to work with the COM position instead of the external forces. Thus, the initial value problem would disappear given that no double integration would be used. Moreover, it should be easier to control the shape of the trajectory (instead of forces) while imposing initial and final position and velocity. Nevertheless, it should also be necessary to take the shape of external forces into account in order to preserve dynamics involved in the original motion. The method proposed in this paper still has the advantage of directly controlling the forces and should naturally take new dynamic constraints into account (such as taking wind, changes in gravity, changes in mass repartition... into account), contrary to approaches only based on kinematics.

Several methods were proposed in biomechanics, computer animation and robotics to model vertical jumps. However, there are many different ways to perform a jump, according to styles and performance (in long jump compared to smash in volley-ball for example). Modelling all the possible styles is quite impossible. However, searching how to adapt a captured motion to new constraints is supposed to preserve the original style and looks more promising, as suggested for example [15]. The applications are numerous, ranging from computer animation to sports motion and performance understanding.

Acknowledgements

References

1. Witkin, A., Kass, M.: Spacetime constraints. In: Proceedings of ACM SIGGRAPH, Atlanta, Georgia, Addison Wesley (1988) 159–168
2. Cohen, M.: Interactive spacetime control for animation. Proceedings of ACM SIGGRAPH '92 **.26** (1992) 293–302 Chicago, Illinois.
3. Liu, Z., Gorther, S., Cohen, M.: Hierarchical spacetime control. Proceedings of ACM SIGGRAPH '94 (1994) 35–42 Orlando, Floride.

4. Gleicher, M.: Retargetting motion to new characters. In: Proc. of ACM SIG-GRAPH. (1998) 33–42

5. Lee, J., Shin, S.: A hierarchical approach to interactive motion editing for human-like figures. Proceedings of ACM SIGGRAPH 99 (1999) 39–48

6. Ménardais, S., Multon, F., Kulpa, R., Arnaldi, B.: Motion blending for real-time animation while accounting for the environment. In: Computer Graphics International. (2004)

7. Shin, H., Kovar, L., Gleicher, M.: Physical touch-up of human motions. In: Proceedings of Pacific Graphics, Alberta, Canada (2003)

8. Boulic, R., Fua, P., Herda, L., Silaghi, M., Monzani, J., Nedel, L., Thalmann, D.: An anatomic human body for motion capture. In: Proc. of EMMSEC'98, Bordeaux, France (1998)

9. LeCallennec, B., Boulic, R.: Interactive motion deformation with prioritized constraints. In R. Boulic, D.P., ed.: Proceedings of ACM/Eurographics SCA, Grenoble, France (2004) 163–171

10. Baerlocher, P., Boulic, R.: An inverse kinematic architecture enforcing on arbitrary number of strict priority levels. The Visual Computer **20** (2004) 402–417

11. Tak, S., Song, O., Ko, H.: Spacetime sweeping: a interactive dynamic constraints solver. In: Proceedings of IEEE Computer Animation. (2002) 261–270

12. Hodgins, J.: Animating human athletics. Proceeding of ACM SIGGRAPH'95 (1995) Los Angeles, California.

13. Fang, A., Pollard, N.: Efficient synthesis of physically valid human motion. In: Proceedings of ACM SIGGRAPH. Volume 22. (2003) 417–426

14. Hodgins, J., Pollard, N.: Adapting simulated behaviors for new characters. In: Proceedings of ACM SIGGRAPH, Los Angeles, California, Addison Wesley (1997)

15. Pollard, N., Behmaram-Mosavat, F.: Force-based motion editing for locomotion tasks. In: Proceedings of IEEE international conference on Robotics and Automation, San Francisco, CA (2000)

16. Sherwood, D.: Impulse characteristics in rapid movement : implications for impulse-variability models. Journal of motor behavior **18** (1986) 188–214

17. Herda, L., Fua, P., Plänkers, R., Boulic, R., Thalmann, D.: Using skeleton-based tracking to increase the reliability of optical motion capture. Human Movement Science Journal (2001)

18. DeLeva, P.: Adjustements to zatsiorsky-seluyanov's segment inertia parameters. Journal of Biomechanics **29** (1996) 1223–1230

19. Pearsall, D., Costigan, P.: The effect of segment parameter error on gait analysis results. Gait and Posture **9** (1999) 173–183

20. Vaughan, C., Andrews, J., Hay, J.: Selection of body segment parameters by optimization methods. Journal of biomechanical engineering **104** (1982) 38–44

21. Liegeois, A.: Automatic supervisory control of the configuration and behavior of multibody mechanisms. IEEE Trans. Systems, Man, and Cybernetics **7** (1977) 868–871

Upper-Limb Posture Definition During Grasping with Task and Environment Constraints

Nasser Rezzoug and Philippe Gorce

LESP EA 3162, Université du Sud Toulon-Var,
avenue de l'université BP 20132,
83957 La Garde Cedex, France
{gorce, rezzoug}@univ-tln.fr

Abstract. The purpose of this study is to propose a new tool to define the posture of a complete upper-limb model during grasping taking into account task and environment constraints. The developed model is based on a neural network architecture mixing both supervised and reinforcement learning. The task constraints are materialized by target points to be reached by the fingertips on the surface of the object to be grasped while environment constraints are represented by obstacles. Without few prior information on the adequate posture, the model is able to find a suitable solution. Simulation results are proposed and commented. This tool can find interesting applications in the frame of gesture definition and simulation.

1 Introduction

Defining a suitable upper-limb configuration to perform a manual gesture (e.g. grasping) is a challenging problem especially in an environment with obstacles. Indeed, the completion of this task implies the consideration of a large number of constraints related not only to the structure of the limb and the characteristics of the object but also to the requirements of the task and the state of the environment. In this frame, a new architecture based on neural networks is proposed to define the kinematic configuration of a 27 degrees of freedom upper-limb model from the knowledge of the position of the object to be grasped, the contact set and the bounds of the 7D arm configuration space. No information about the obstacles is available to the learning agent. Unlike previous developed tools [1 - 4], the proposed method can take into account both task and environment constraints in a straightforward way. Also, it is build upon a previous work [5, 6] where only the hand was taken into account with no obstacles in the environment. The consideration of the whole upper-limb including the lower and the upper-arm as well as the presence of obstacles in the environment are new functionalities that enhance the original model performances.

2 Problem Definition and Hypotheses

The definition of the problem is : *Given an object located in an environment with obstacles, define all the kinematic parameters of an upper-limb model (including arm*

and hand) in such a way that the fingertips can reach a defined position on the surface of the object with no collision between any part of the upper-limb and the environment. In the remainder of this article, the term posture refers to the set of joint angles that describes the upper-limb model configuration. This includes the arm (7 degrees of freedom) as well as the hand and the fingers (4 degrees of freedom per finger). The number of contacts is between two and five. To define the upper-limb posture, the following assumptions are made :

1. Only precision grasps are considered, therefore there is only one contact per finger,
2. We assume that the contact set between the fingers and the object is already defined and satisfies force closure (i.e. the forces applied by the fingers can ensure the object immobility and equilibrium under frictional constraints),
3. To each contact on the object corresponds a particular finger.

The input data of the model are:

1. The upper limb geometry,
2. The number of contacts,
3. The finger associated to each contact,
4. The object location and orientation,
5. The location of the contacts in the object reference frame,
6. The bounds of a seven dimensional search space corresponding to the set of possible arm configurations.

   The outputs of the model are:

1. The configuration of the arm in joint coordinate space,
2. The location and orientation of the hand reference frame relative to the world frame,
3. The configuration of the fingers in joint coordinate space.

   No assumption is made about the number, shape, position and orientation of the obstacles.

2.1 Hand Model

The hand model is composed of five articulated rigid chains representing the fingers. They are connected to a common body representing the palm (Fig. 1). Each finger has three links connected by three joints with a total of four degrees of freedom. The first joint of each finger has two degrees of freedom that allow to simulate the flexion-extension and abduction-adduction movements. The two other joints have one degree of freedom representing joint flexion. The complete model has 20 degrees of freedom. The hand geometry is set according to studies on hand anthropometry carried out by Garret [7] and Buchholz and al. [8].

2.2 Arm Model

The model of the arm is composed of two segments and three joints (Fig. 1). The first joint, located at the shoulder (gleno-humeral joint) has three degrees of freedom (ball

and socket joint with 3 rotations of an amount q_1, q_2 and q_3). The second joint is located at the elbow and has one degree of freedom (flexion of an amount q_4). Finally, the last joint, located at the wrist, has three degrees of freedom (rotation of an amount q_5, q_6 and q_7). The final frame of the last segment defines the orientation of the hand palm. We consider that the joint responsible for wrist axial rotation is located at the wrist. According to this formulation, the arm posture is completely defined by the joint angles vector $\mathbf{q} = [q_1, q_2, q_3, q_4, q_5, q_6, q_7]^T$. It can be noted that the arm has one redundant degree of freedom.
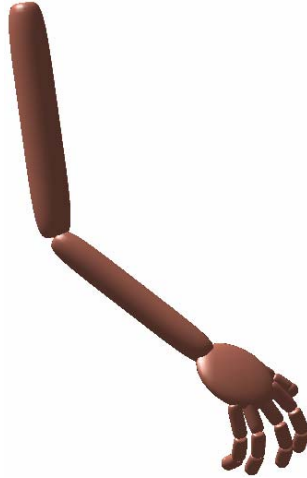
Fig. 1. Upper-limb model

3   Model Presentation

Defining the upper-limb kinematic parameters is a complex task because of the large number of degrees of freedom and constraints to be satisfied. It can be considered as an inverse kinematics problem of a redundant multichain mechanism with multiple objectives characterized by a possibly infinite number of solutions. To solve this problem, we start from the idea that if the hand palm configuration is held fixed it is possible to compute the configuration of the fingers in joint coordinate space by using inverse kinematics. Furthermore, if the desired fingertip position is expressed relative to the finger root frame and if a model of finger inverse kinematics is constructed by learning, it is possible to compute quickly the fingers joint angles given any hand palm configuration. In this way, the number of parameters to define decreases from up to 27 to 7 which correspond to the arm joints configuration. This latter remains to be computed. The chosen mechanism uses reinforcement learning to optimize the arm configuration such that the fingertips can reach the contact on the surface of the object. In order to induce a collision avoidance behavior specific reinforcement signals are proposed and evaluated. In particular, a simple but

efficient mechanism based on the concept of shaping improves the learning performances. The process of upper-limb grasping posture definition has two parts: the arm joints configuration is generated by a first network (called "Arm Configuration Neural Network" or ACNN) and the corresponding finger joints are obtained by inverse kinematics with a second neural network model called "Finger Configuration Neural Network" or FCNN. Then the whole posture is evaluated and from the corresponding evaluation the ACNN is trained by reinforcement learning in order to increase the performance over time.

The model is composed of two neural modules working in closed loop (Fig. 2). The first module (ACNN) is aimed at determining the appropriate arm joints configuration. It is a multilayer feedforward neural network with a learning process based on the paradigm of reinforcement learning. This type of learning is achieved with resorting to a particular type of processing units called Stochastic Real Valued (SRV) neurons [9]. The FCNN, the second module, is devoted to the definition of the fingers configuration in joint coordinate space from the desired fingertips position [5, 6, 10]. Its output data are used to evaluate the arm configuration and compute a reinforcement signal taking values over the interval [0, 1]. This evaluation of the current upper limb configuration is used by the first network (ACNN) to update its internal parameters. The principle of the developed method is now described:
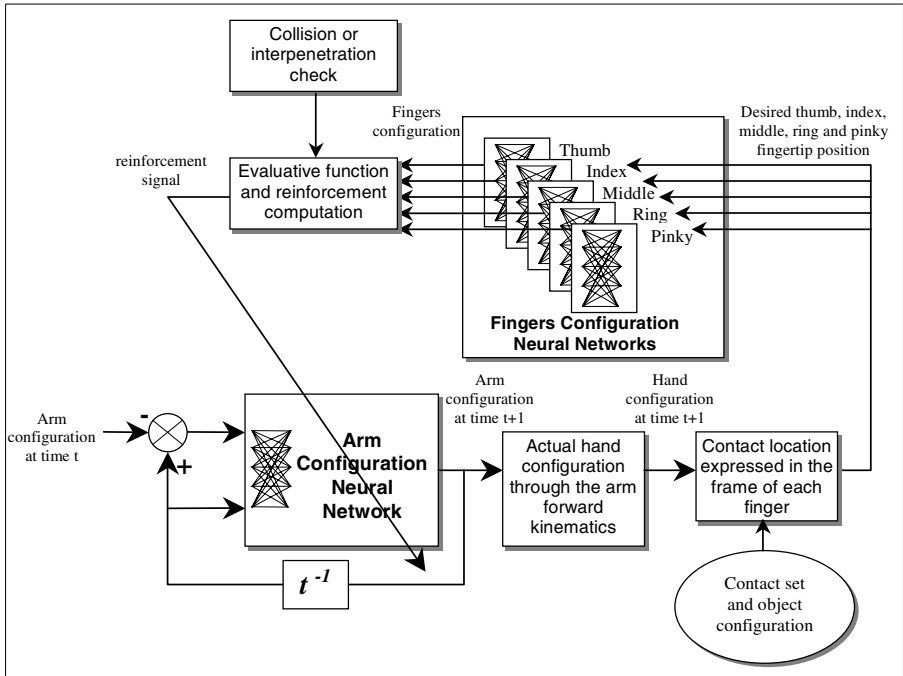
Fig. 2. Model structure

1. The current arm configuration and the difference between the actual and previous arm configurations (equivalent to a speed with unit time) are input to the ACNN. As output, a new arm configuration within the defined search space bounds is obtained.
2. From the arm configuration, the hand palm frame configuration (position and orientation) is computed by using the arm forward kinematics.
3. Thanks to this new hand palm configuration, it is possible to express the position of the contacts in the frame of each corresponding finger (on the surface of the object, a particular contact is affected to each finger),
4. Using the inverse kinematics scheme (FCNN), the joint configuration of the fingers is computed.
5. The complete upper-limb configuration (arm joints angles, hand palm configuration and fingers joint angles) is tested with a criterion. From this latter, a reinforcement signal is computed and is used to update the ACNN internal parameters.

This procedure is repeated until a good solution is found (i.e. the reinforcement reaches a sufficiently high value and the criterion is optimized) or until the maximum number of iterations is reached. More details about the FCNN and ACNN can be found in [5, 6]. In these references, only the hand is considered. Therefore we use the term HCNN (Hand Configuration Neural Network) instead of ACNN.

4 Improving Learning Performances by Shaping

In order to define a suitable reinforcement signal, two aspects of the performance have to be taken into account. The first one evaluates the upper-limb positioning task while the second is relative to collision avoidance. In the following, the different steps that conduct to the definition of the appropriate reinforcement signal are described. Firstly, the positioning task is evaluated. To do this, given the arm and fingers configurations, the actual position of the fingertips is calculated using forward kinematics.

$$\text{Let } \mathbf{PX}_i^D = \left(x_i^D, \, y_i^D, \, z_i^D \right)^T \tag{1}$$

be the vector of the desired fingertip position written relative to the base coordinate frame of finger i at step k and

$$\mathbf{PX}_i^M = \left(x_i^M, \, y_i^M, \, z_i^M \right)^T \tag{2}$$

the vector of the actual fingertip position written relative to the base coordinate frame of finger i. If n fingers are involved, the total error at step k is:

$$E_k = \sum_{i=1}^n \left\| \mathbf{PX}_i^D - \mathbf{PX}_i^M \right\| \tag{3}$$

with $\left\| . \right\|$ designing the Euclidean L2 norm.

The simplest form of the reinforcement R_1 as used in [5, 6] gives a maximum penalty if error E_k is large and is given in (4):

$$R_1 = 1 - h\left(a.E_k\right).$$

(4)

where a is a positive real number.

The function h is chosen in such a way that R_1 is a decreasing function of the error E_k and takes values over the interval [0, 1]. If E_k is large, h tends toward 1 and therefore the network receives a maximum punishment with a reinforcement R_1 toward 0. On the other hand, if the error E_k is low, h tends toward 0 and consequently the system receives a reinforcement toward 1. In the present case, h is the tangent sigmoid function.

Starting from the definition of R_1, the basic expression of the reinforcement signal that incorporates collision avoidance behavior is given by:

$$R_2 = \begin{cases} R_1 & \text{if no collision} \\ R_1/2 & \text{if collision} \end{cases}$$

(5)

In order to fulfill the secondary task i.e. collision avoidance, the reinforcement R_1 is divided by two whenever a collision is detected. Therefore, even if the principal task is accomplished with success the reinforcement is low due to the occurrence of a collision. One can notice the simplicity of the incorporation of collision avoidance behavior in the learning process. However, the criterion R_2 uses a somewhat crude strategy and the results may not be as satisfying as expected. Indeed, the learning agent has to directly discover the right strategy to satisfy two kinds of constraints at the same time. This is a more complex task than arm positioning only.

In order to circumvent this difficulty, we propose to use a technique inspired from animal training called shaping [11]. Gullapalli [12] gave a nice definition of this concept and applied it to the frame of reinforcement learning : "The principle underlying shaping is that learning to solve complex problems can be facilitated by first learning to solve simpler problems. … the behavior of a controller can be shaped over time by gradually increasing the complexity of the task as the controller learns".

To incorporate shaping in the learning procedure, the basic idea is to let the agent learn the positioning task first and the collision avoidance behavior during a second phase. To implement this, a reinforcement signal that gradually increases over time the penalty due to collisions is defined. In this way, the agent can learn adequately the first task and modify its behavior in order to achieve the second one. The reinforcement value used in this case is the following:

$$R_3 = \begin{cases} R_1 & \text{if no collision} \\ R_1/\left(1+i/p\right) & \text{if collision} \end{cases}$$

(6)

where i is the current iteration number and p the maximum number of iterations.
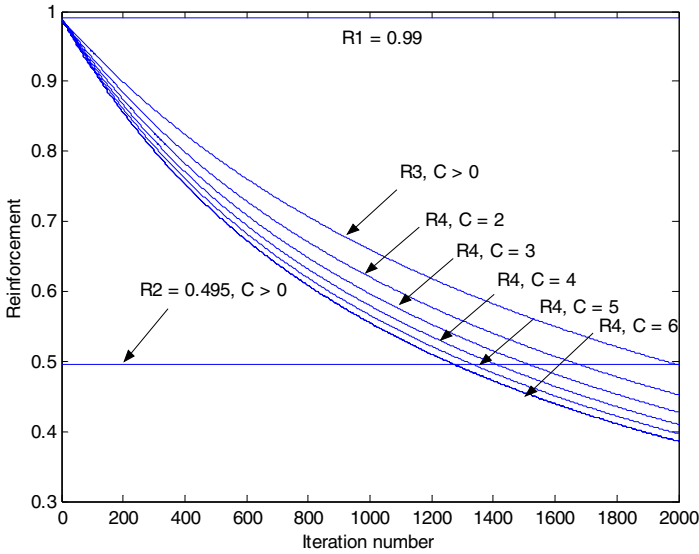
Fig. 3. Evolution of the reinforcements R_1, R_2, R_3 and R_4 relative to the iteration and collision numbers

If collisions occur, for the same value of R_1, an increase of i conducts to an increase of the denominator in (6) and consequently to a decrease of R_3. If $i = p$, we can notice that $R_3 = R_2$ and that there is a gradual shift from R_1 (no penalty for collision) to R_2 (full penalty for collision). This weaker definition of arm positioning with collision avoidance may be easier to learn than direct collision avoidance as defined by R_2. The evolution of R_3 with $R_1 = 0.99$ when collisions occur is displayed in Fig. 3.

The main drawback of R_3 is that the same penalty is applied whatever the number of collisions. It may be easier to learn the task successfully if the learning agent can grade differently two situations with different numbers of collision, giving more penalty to the posture conducting to more collisions or interpenetrations. In order to solve this problem, we define the reinforcement R_4:

$$R_4 = \begin{cases} R_1 & \text{if no collision} \\ R_1 / \left(1 + c^\beta \left(i / p\right)\right) & \text{if collision} \end{cases} \tag{7}$$

where c is the number of detected collision(s) or interpenetration(s) and β a positive real number.

Reinforcements R_3 and R_4 use the same strategy, except that R_4 takes into account the number of collisions. Indeed, for the same value of R_1, i and p, an increase of c conducts to an increase of the denominator in (7) and therefore to a decrease of the reinforcement R_4. If $c = 1$, we notice that $R_4 = R_3$. The evolution of R_4, with different values of c is displayed in Fig. 3.

5   Simulation Results

The task to be performed is to grasp a cylinder with three fingers. Two different environments are considered, the first one with a big obstacle between the arm and the object and the second one with two obstacles (Fig. 4.). 30 simulations are performed for each reinforcement and for each environment. The weights of the ACNN are initialized with random values over the interval [-0.5, 0.5] and a random arm configuration is chosen within the search space. The learning has to be completed for each task and environment and is performed until a reinforcement greater than 0.99 is obtained or until the maximum number of iterations is reached. A FCNN is constructed off line for each finger before the simulations [5, 6]. Collision or interpenetration check is implemented with a two steps scheme. Axis aligned bounding boxes are constructed for each element of the environment to make a first check. If it is positive, the distance between any pairs of solids that are likely to collide is computed. This is done by minimizing the distance between any pair of points on the surface of two elements of the scene modelled with superquadrics.
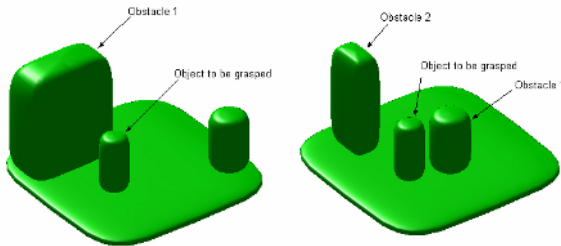
Fig. 4. Environments for the two grasping tasks

In table 1, we display the obtained results. In the first row, the number of successes is indicated for each reinforcement. This corresponds to the case where the reinforcement is greater than 0.99. In the second and third rows is indicated the number of cases for which a failure is due either to the positioning task or to collisions. Finally, for the successful cases, the last two rows indicate the mean and standard deviation of the required number of iterations to obtain a suitable reinforcement.

Reinforcement R_1 is used as a reference in order to demonstrate that the other reinforcements R_2, R_3 and R_4 have effectively an effect on collision avoidance.

The first observation is that the incorporation of collision avoidance behaviour in the reinforcement signal effectively leads to collision avoidance even if the positioning task is not achieved. Using R_1, we obtain 22 solutions out of 26 valid ones with collisions between the upper limb and the environment for the first task and 24 out of 25 for the second task. This number falls to 3, 7 and 4 for R_2, R_3 and R_4 for the

Table 1. Simulation results for task 1

Reinforcement		R_1	R_2	R_3	R_4
Success		26	20	22	26
Causes of failure	Positioning task	4	10	7	4
	Collision	22	3	6	2
Mean iterations number		102	430	281	310
Standard deviation		126	271	259	245

Table 2. Simulation results for task 2

Reinforcement		R_1	R_2	R_3	R_4
Success		25	8	22	16
Causes of failure	Positioning task	5	22	7	14
	Collision	24	4	2	5
Mean iterations number		120	454	273	260
Standard deviation		161	266	228	223

first task and 4, 2 and 5 for the second task respectively. Also, we notice that there is an increase of the number of successes when shaping is used compared to the case where a crude collision avoidance reinforcement is used (R_2). This is particularly obvious for the second task (8 successes with R_2 compared to 22 using R_3). This suggests that the strategy of shaping allows to find a solution more often and therefore that it facilitates the learning. To determine if the use of the different reinforcements has an effect on the number of iterations (NOI), a one way analysis of variance (ANOVA) [13, 14] on the number of iterations to complete the task is conducted. A Bonferoni post-hoc test is used to perform multiple comparisons between means. The ANOVA evidences a significant difference between four groups means ($p<0.0001$). Also, the post-hoc tests show a significant increase of the NOI using R_2 compared to the NOI using R_3 and R_4 ($p<0.05$). Also, a significant increase of the NOI using R_2, R_3 and R_4 compared to the NOI using R_1 is noticed ($p<0.05$). There is no significant difference between the NOI using R_3 and R_4. These results suggest that learning the positioning task is easier than the positioning task with collision avoidance because, on average, more iterations are needed whatever the chosen reinforcement. Secondly, the incorporation of shaping in the learning process reduces significantly the required number of iterations to reach the goal. Finally, taking into account the number of collisions in the reinforcement definition does not seem to improve significantly the learning performances. Therefore, among all the reinforcement signals proposed in this study, we can consider that R_3 is the best one to perform grasping posture definition with obstacles in the frame of the considered model.

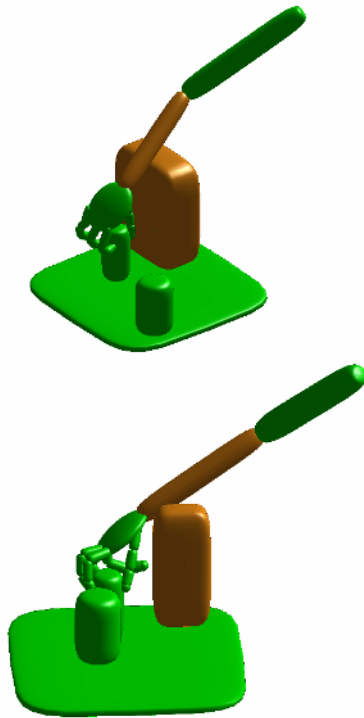In Fig. 5 examples of postures obtained with R_3 for the two tasks are displayed.

Fig. 5. Postures obtained with R_3 for the two tasks

Also, in Fig. 6, the posture obtained to grasp a cylinder surrounded by 3 obstacles is shown.
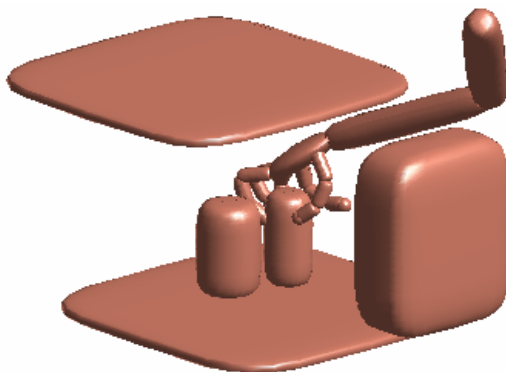
Fig. 6. Upper-limb posture obtained by the model to grasp a cylinder surrounded by three obstacles

6   Conclusion

In this paper, we have proposed a new model to define the kinematics of an upper limb model during grasping. The proposed method is based on two neural networks. The first one is dedicated to finger inverse kinematics. It is based on an architecture composed of several networks. The second stage of the model uses reinforcement learning to define the appropriate arm configuration. The corresponding neural network is composed of backpropagation units associated with stochastic real valued (SRV) neurons in the output layer. This model is able to define the whole upper limb configuration to grasp an object while avoiding obstacles located in the environment. Several simulation results demonstrate the capability of the model. The fact that no candidate solution is required to start the upper-limb posture construction is an interesting property of this method. Another valuable feature is that a solution can be obtained after a relatively low number of iterations and that no information about the number, position, shape and size of the obstacles is provided to the learning agent. We can consider this method as a part of a larger model to define arm postures that tackles the "kinematic part" of the problem and can be associated with any grasp synthesis algorithm. In future work, we plan to develop algorithms based on unsupervised learning and Hopfield networks to construct the upper-limb movement. In this way, we will be able to generate an upper-limb collision free trajectory in joint coordinate space from any initial position to the collision free final configuration obtained by the method described in this article.

References

1. Maurel, W., Thalmann, D. : Human upper limb modeling including scapulo-thoracic constraints and joint sinus cones, Computers and Graphics 24:2 (2000) 203-218
2. Tolani, D., Goswami, A., Badler, N. : Real-time inverse kinematics techniques for anthropomorphic limbs, Graphical Models 62:5 (2000) 353-388
3. Wang, X. : A behavior-based inverse kinematics algorithm to predict arm prehension postures for computer- aided ergonomic evaluation, J. Biomech. 32 (1999) 453-460
4. Wang, X., Verriest, J. P. : A geometric algorithm to predict the arm reach posture for computed–aided ergonomic evaluation, J. of Visualization and Computer Animation 9 (1998) 33-47
5. Rezzoug, N., Gorce, P. : A biocybernetic method to learn hand grasping posture, Kybernetes 32:4 (2003) 478-490
6. Gorce, P., Rezzoug, N. : A method to learn hand grasping posture from noisy sensing information, Robotica 22:3 (2004) 309-318
7. Garret, J. W. : The adult human hand : some anthropometric and biomechanical considerations, Human factors 13 (1971) 117-131
8. Buchholz, B., Armstrong, T. J., Goldstein, S. A. : Anthropometric data for describing the kinematics of the human hand, Ergonomics 35 (1992) 261-273
9. Gullapalli, V. : A stochastic reinforcement learning algorithm for learning real valued functions, Neural Networks 3 (1990) 671-692

10. Oyama, E., Agah, A., MacDorman, K. F., Maeda, T., Tachi, S. : A modular neural network architecture for inverse kinematics model learning, Neurocomputing 38-40 (2001) 797-805
11. Skinner, B.F. : The behavior of organisms : An experimental analysis. D Appleton century, New York (1938)
12. Gullapalli, V. : Reinforcement learning and its application to control, PhD Thesis, University of Massachusetts (1992)
13. Fisher, R. A. : The logic of inductive inference, Journal of the Royal Statistical Society, 98 (1935) 39-82
14. Miller, R.G. : Beyond Anova, Basics of applied statistics. Chapman & Hall/CRC, Boca Raton FL (1997)

Adaptive Sampling of Motion Trajectories for Discrete Task-Based Analysis and Synthesis of Gesture

Pierre-François Marteau and Sylvie Gibet

Valoria, Université de Bretagne Sud, Campus de Tohannic, rue Yves Mainguy,
F-56000 Vannes, France
`{Pierre-Francois.Marteau, Sylvie.Gibet}@univ-ubs.fr`

Abstract. This paper addresses the problem of synthesizing in real time the motion of realistic virtual characters with a physics-based model from the analysis of human motion data. The synthesis is achieved by computing the motion equations of a dynamical model controlled by a sensory motor feedback loop with a non-parametric learning approach. The analysis is directly applied on end-effector trajectories captured from human motion. We have developed a Dynamic Programming Piecewise Linear Approximation model (*DPPLA*) that generates the discretization of these 3D Cartesian trajectories. The *DPPLA* algorithm leads to the identification of discrete target-patterns that constitute an adaptive sampling of the initial end-point trajectory. These sequences of samples non uniformly distributed along the trajectory are used as input of our sensory motor system. The synthesis of motion is illustrated on a dynamical model of a hand-arm system, each arm being represented by seven degrees of freedom. We show that the algorithm works on multi-dimensional variables and reduces the information flow at the command level with a good compression rate, thus providing a technique for motion data indexing and retrieval. Furthermore, the adaptive sampling seems to be correlated with some invariant law of human motion.

1 Introduction

When simulating and animating virtual characters, biologically-inspired models play a major role, as the produced movements exhibit properties inherent to human movements. As emphasized by psychologists and physiologists, human beings are more sensitive to movement of biological origin [1]. There are two ways to reach a certain degree of naturalness for the synthesis of gestures. First, the articulated system can be modeled by a physical model responding to physical laws of movement. The difficulty here is not so much to simulate the motion equations, but to determine the appropriate controller that drives the system towards a desired goal expressed in the task-command space. Second, the animation can be directly done by motion capture data. The difficulty with this last method is to determine new motion on the basis of previously registered elementary motions, and to ensure smooth transitions between these elementary motions.

In our approach, we try to establish a link between real-time synthesis models and motion analysis models using motion capture data. The synthesis model is based on a dynamical Sensory Motor Model (*SMM*) and is driven by a task-based analysis model that extracts discrete target-based patterns on the basis of motion capture data.

After presenting the global analysis synthesis method, this paper is focused on the presentation of the analysis model. More precisely we describe a Dynamic Programming Piecewise Linear Approximation (*DPPLA*) algorithm whose outputs can be used to control the articulated system. Since the command is directly extracted from real human data, the animation system produces movement with biological relevance. The analysis-synthesis model is applied to the simulation of anthropomorphic hand-arm movements. The performance of the method is presented according to invariant laws of movement.

2   Analysis-Synthesis Methods

Several problems arise when simulating human hand-arm motion. First, the biomechanical system composed of a set of interacting articulated structures has to be has to be modeled. This biomechanical model is directly dependent on the way the muscular-skeleton apparatus is modeled. Most of the time simplifications of the mechanical structure have to be considered. Controlling such a complex system necessitates the design of appropriate controllers associated to the different articulated chains. In our approach, the control of the biomechanical system is materialized by a Sensory Motor Model (*SMM*) that continuously uses sensory data to update the state variables of the dynamical system to control. The feedback mechanism carries out an inversion process, i.e. it automatically computes the input of the biomechanical system from the observable outputs and the command input. The *SMM* is illustrated in Figure 1, as the *motion synthesis* block.
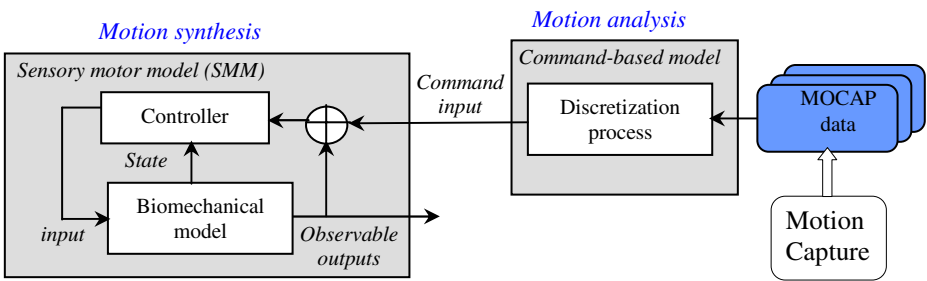
Fig. 1. Analysis/synthesis model for gesture modeling and animation

This paper deals with the problem of modeling the command of such a sensory motor model, as a *motion analysis* approach. This analysis process can be seen as an inversion process: from motion capture signals, a discrete command pattern is extracted through a discretization process; the command input can be used to control the *SMM*, as illustrated in Figure 1.

2.1   Motion Analysis

Exploiting analysis data to run the synthesis model raises the question of identifying the appropriate variables at the command level. Some psychologists and physiologists

assume that - at least for some classes of movements - spatial representation is more invariant than force-time patterns or joint rotations. Following this idea, we make the hypothesis that end-effectors motion traces expressed in the 3D Cartesian space can be used to control the muscular-skeleton system. We expect that this spatial representation is closer to the task than other internal sensory or motor variables, as force or moment variables [2-3].

More precisely, this paper will cover the modeling of the end-effectors trajectories in terms of discrete target patterns. Given a desired trajectory, we extract sequences of targets that represent in an optimal way the original trajectory. This approach differs from the methods widely used for the purpose of segmenting human motion capture data into high-level behaviors or low-level components. These last methods are designed most of the time to reduce dimensionality by identifying low-dimensional clusters in high-dimensional data [4]. Another method is proposed for non-uniform subsampling of motion captured data. This method uses polygonal approximation to provide a compressed representation of dance gesture trajectories [5]. The objectives of this approach are different from ours, since the compressed data in [5] are used as input of a recognition system, whereas we use our reduced trajectories for synthesis purpose.

Our target-based discretization method, also called *adaptive sampling* could be useful for motion segmentation, but above all we aim to extract discrete patterns as input of our motion generation models. The main interest of the adaptive sampling is to facilitate the manipulation (edition, recombination) of elementary movements, the composition operators being implemented as the concatenation and smoothing of target sequences. Furthermore this discretization process leads to the reduction of the data flow at the command level and to the reduction of the representation space in which movements are embedded. This is also of great importance for dealing with information retrieval in movement data base. Finally this sampling process is a first step towards the parameterization of motion. Rather than using straightforward key-points extracted from motion invariant laws [6], we propose an automatic segmentation process operating on multi-channel variables which yields a discrete representation of motion.

2.2   Motion Synthesis

Numerous solutions exist to control sensory motor systems. Some methods consider the control problem as finding numerical solutions to inverse kinematics or inverse dynamics, depending on the representation of the movement system. Among these solutions, we have developed analytical methods extended by learning methods, applied both for kinematics or dynamics control.

The synthesis model attached to each articulated chain is represented by a sensory motor system which yields a means of coupling the continuous internal signals in the sensory motor process with desired commands expressed as continuous trajectories in the Cartesian space or as sequences of discrete targets in the task space. It has already been developed for controlling various articulated systems with different control policies: The *GSM* model uses a gradient-based algorithm in a sensory motor closed-loop transformation which integrates neurophysiological elements [7]. This model has proved to control articulated chains and produce motion that globally respects human motion laws. It has been used in a modular architecture to generate

expressive communicative and Sign Language Gestures [8-9] or coordinated juggling motion [10].

Another control policy uses a learning approach within a sensory motor loop (*ASM model*) [11-12]. In this case, the learning algorithm is based on a local inversion principle and uses local neighborhood research techniques to compute the new predicted state variation to be generated. The training data set can be incrementally updated in time to adapt to changes of the performance tasks and to structural changes of the physical system.

A sensory motor dynamical system is considered in this paper, which includes two inversion processes: the first one is a kinematics inversion, based on the same learning control policy; the second one is a dynamics inversion, achieved by a Proportional Integrative Derivative model.

For each sensory motor model, various tasks can be defined, such as reaching or tracking tasks. In our synthesis models, the task is expressed as desired goals in the spatial 3D space or in the joint angular space. For simple or multiple reaching tasks, these goals can be represented as sequences of targets to reach, with or without co-articulation. For tracking tasks, the goals can be expressed as desired end-point continuous or discrete trajectories.

We propose to define the input command from the analysis of motion capture data. Coupling synthesis with data extracted from human motion is necessary if we want to integrate some invariant features of movements. The discretization method, called Dynamic Programming Piecewise Linear Approximation (*DPPLA*), achieves the analysis process necessary for the synthesis process, which was missing in our former systems.

3   Dynamic Programming Piecewise Linear Approximation Model (*DPPLA*)

The *DPPLA* algorithm makes the discretization of end-effector trajectories possible in $O(n^2/k)$ where k is the number of samples. These trajectories can be considered as a multivariate continuous *3D* process $X(t) = [x(t), y(t), z(t)]$. A more general view is to consider $X(t)$ as a spatio-temporal trajectory of time-stamped spatial vectors in p dimensions. In practice, we will deal with the sampled trajectory $X(n)$ where n is the time-stamp index.

We propose a data modeling approach to handle the adaptive sampling of the end-effector trajectories. More precisely, we are seeking an approximation $X_{\hat{\theta}}$ of $X(n)$ such as:

$$\hat{\theta} = \underset{\theta}{ArgMin}\big(E(X, X_{\theta})\big)$$

where E is the *RMS* error between X and the model X_{θ}.

As a first attempt, we have selected the family $\{ X_{\theta}(n) \}$ as the set of piecewise linear functions. Numerous methods have been proposed to the problem of approximating multidimensional curves using piecewise linear simplification and dynamic programming in $O(kn^2)$ complexity [13]. Some efficient algorithms [14] (in $O(nlog(n))$ complexity) [15] have been proposed for planar curves, but none for the

general case in R^d. We constraint the search of the segments by imposing that the extremities of the piecewise linear segments are on the trajectory $X(t)$. Thus, θ is the set of discrete time location $\{n_i\}$ of the segments' endpoints. Since the end of a segment is the beginning of the following one, two successive segments share a common n_i at their interface. The selection of the optimal set of parameters $\hat{\theta} = \{\hat{n}_i\}$ is performed using a dynamic programming algorithm [16] as follows.

We first define the compression rate of the piecewise approximation as:

$$\rho = 1 - \frac{\left|\{n_i\}\right|}{\left|\{X(n)\}\right|} \times \frac{p+1}{p}$$

where $|A|$ stands for cardinal of set A and $X(n) \in \Re^p, \forall n$

Given a value for ρ and the size of the trajectory window to sample $w = \left|\{X(n)\}_{n \in \{1,...,w\}}\right|$, the number $N = \left|\{n_i\}\right| - 1$ of piecewise linear segments is known.

Let us define $\theta(k)$ as the parameters of a piece wise approximation containing k segments, and $\delta(k,i)$ as the minimal error between the best piecewise linear approximation containing k segments and covering the discrete time window $\{1,..,i\}$:

$$\delta(k,i) = \underset{\theta(k)}{Min}\left\{\sum_{n=1}^{i}\left\|X_{\theta(k)}(n) - X(n)\right\|^2\right\}$$

According to the Bellman optimality principle, $\delta(k,i)$ can be decomposed as follows:

$$\delta(k,i) = \underset{n_k \leq i}{Min}\left\{d(n_k,i) + \delta(k-1,n_k)\right\}$$

where $d(n_k,i) = \sum_{n=n_k}^{i}\left\|Y_{k,i}(n) - X(n)\right\|^2$ and $Y_{k,i} = (X(i) - X(n_k)).\dfrac{n-n_k}{i-n_k} + X(n_k)$ is the linear segment between $X(i)$ and $X(n_k)$.

The initialization of the recursion is obtained observing that:

$$\forall k, \forall i < k, \delta(k,i) = 0$$

The end of the recursion gives the optimal piecewise linear approximation, e.g. the set of discrete time locations of the extremity of the linear segments:

$$\hat{\theta}(k) = \underset{\theta(k)}{ArgMin}\left\{\sum_{n=1}^{w}\left\|X_{\theta(k)} - X(n)\right\|^2\right\}$$

with the minimal error :

$$\delta(k,w) = \sum_{n=1}^{w}\left\|X_{\hat{\theta}(k)}(n) - X(n)\right\|^2$$

The complexity of the proposed algorithm is in $O(k.w^2)$. To reduce this complexity, the search window can be limited by using a lower bound factor for each step i: $lb = max\{i-band, 0\}$, where $band$ is a parameter fixed by the user:

$$\delta(k,i) = \underset{lb \le n_k \le i}{Min} \left\{ d(n_k,i) + \delta(k-1,n_k) \right\}$$

In practice we use *band = 2*w/k*, leading to the complexity $O(w^2/k)$.

4   Runtime Synthesis

Here a generic learning model for the control of a dynamical articulated model is used. The control policy associated to this learning method jointly achieves the kinematics and the dynamics inversion of the system. The kinematics inversion is carried out by a non parametric learning algorithm which processes the mapping $(y,\ \delta x) \rightarrow \delta y$, *y* being the state and *x* the output of the system (*ASM* model) [12]. The error signals measured between the sensory output and the task input are used as corrective information to update the torque command of the movement system.

The dynamics inversion is assured by a set of controllers acting on the pair $(\tau, \delta q/dt)$, τ being the forces applied on the joints, and *dq/dt* the angular velocity of the joint rotations. These controllers, classically used in robotics and computer animation and issued from linear control theory use Proportional Integrative Derivative principle (*PID*). For each internal joint, each *PID* controller takes as inputs angular position of the joint and its derivative as well as the desired angular position, and computes the torque output required to produce the desired displacement of the joint as expressed by the following equation:

$$\tau(t) = K_p(\vec{q}_T - \vec{q}) + K_d(\dot{\vec{q}}_T - \dot{\vec{q}}) + K_i \int \vec{q}(t)dt$$

where *q* is the angle of the joint, q_T is the desired angle, K_p, K_d and K_i are the proportional, derivative and integral gains. The effect of the *PID* controller is to eliminate large step changes in the errors, thus smoothing the simulated motion.

5   Experiments

In this section, the results of two experiments are presented. One concerns the analysis process applied on motion capture sequences. The second highlights the synthesis process, using the result of the analysis process as the command input of the dynamical system. The motion data used in our experiments were captured from a VICON optical system at the rate of 110 frames/second. We recorded end-arm movements of about three s duration. Subjects were told to perform about ten different random patterns, varying the kinematics and the shape of the patterns.

The analysis is conducted on 3D Cartesian trajectories of the arm extremity. We consider for hand-arm movements that these trajectories express the trace of the task-based command. The *DPPLA* algorithm is applied on these trajectories with varying compression rates that fix the number of targets on the trajectory. The algorithm segments the trajectory by assigning cut targets along the motion sequence. The objective is not only to detect these cut-points, but also to characterize the distribution of relevant samples along the trajectory.

The results of the *DPPLA* algorithm are illustrated in Figure 2 for a compression rate of 85%. This compression rate is an optimized parameter directly linked to the number of discrete targets extracted by the *DPPLA* algorithm. Figure 2 shows the *x, y* and *z* curves of the end-extremity trajectories, both for the real motion capture data and for the simulated data, after applying the *DPPLA* algorithm. It also indicates the
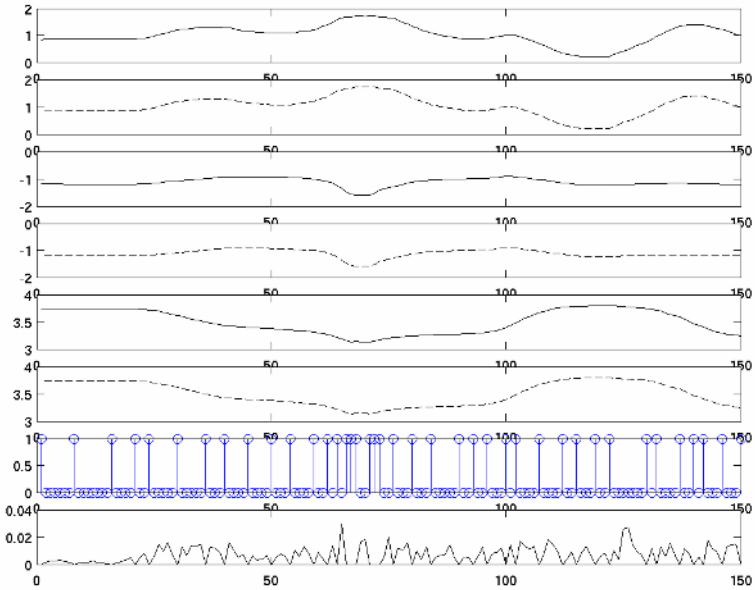
Fig. 2. Three-joint arm simulation with random pattern with a compression rate of 75%; *x, y, z* trajectories: real data *(solid)* and simulated data *(dashed)*; motion separation points assigned by the *DPPLA* algorithm. The *x*-axis corresponds to the frame number, and the vertical bars specify the target points assigned by the algorithm; Reconstruction error.
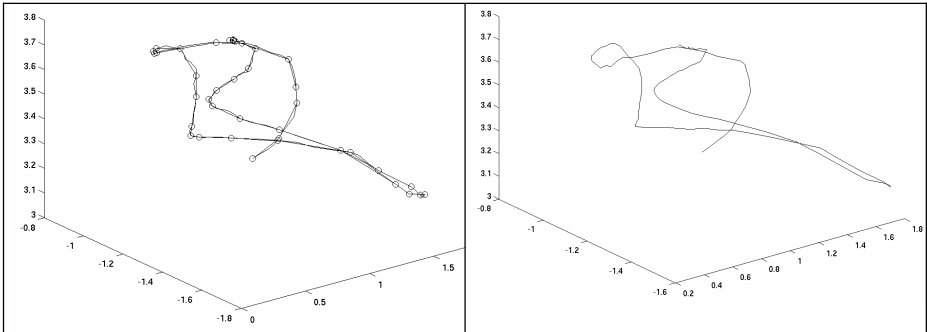
Fig. 3. *(left)* Trajectory of the human wrist in the Cartesian space with the localization of the targets: capture motion data and reconstructed data by linear interpolation; *(right)* Simulated trajectory of the wrist of the virtual dynamical humanoid

location along the frames of the cut-points (targets) that approximate the desired trajectory and the corresponding absolute error between the real data and the simulated ones.

Figure 3 *(left)* gives the shape of the real and reconstructed trajectories from the location of the targets. Some attraction areas where the targets are more concentrated can be seen in this figure: they correspond to zones of larger complexity of the signal. The segmentation induced by the *DPPLA* algorithm can be considered as measured by the density of targets along the motion.

Figure 3 *(right)* illustrates the simulated trajectory performed by the virtual character. Some overshoots can be seen when the curvature is high, due to the fact that the dynamical parameters cannot be adjusted during the course of the simulated motion.

These results demonstrate the tendency of *DPPLA* to increase the target density when the curvature increases. This correlation can be illustrated in Figure 4 with the superposition of the density function representing the spatial concentration of the targets along the motion frames, and the curvature function. Table 1 gives correlation factors between the two functions for three compression rates using a linear regression method.
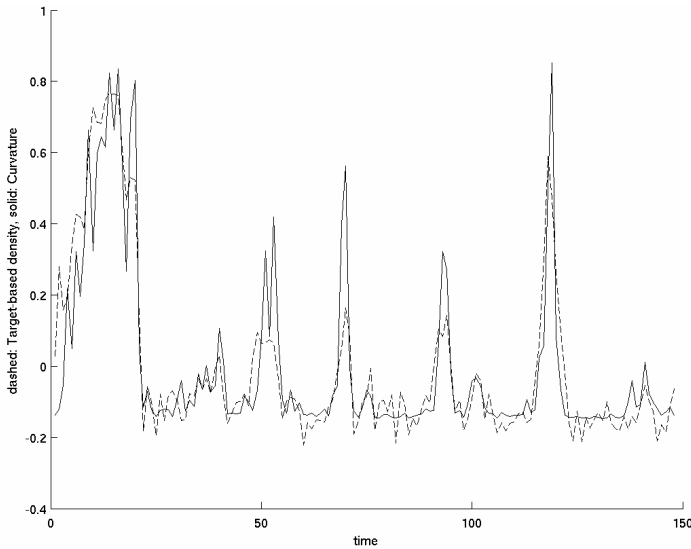
Fig. 4. *(dashed)* Target-based density evolving with time; *(solid)* Normalized curvature evolving with time. The areas where Target-density function is high correspond to areas with high curvature; this function segments the end-point trajectory.

Table 1. Correlation factors between target-based density along the trajectory and curvature for different compression rates

Compression rate	Correlation factor
65%	0.90
75%	0.88
85%	0.84

The *DPPLA* algorithm provides a way to adaptively sample the end-effector trajectory according to the variations of curvature along the trajectory. The analogy between the spatial density of targets and the curvature leads to the investigation of the density function behavior compared with the well-known "two-third power law" [17]. The latter is equivalently expressed by a *one-third power law* relating tangential velocity $v(t)$ to radius of curvature $r(t)$:

$$v(t) = k\, r(t)^{1/3}$$

This law, considered as a basic invariant characteristic of movement, has been demonstrated to be robust for 2D handwriting movements, and also for 3-D elliptical patterns. Figure 5 *(up-left)* illustrates this one-third power law for a random end-effector trajectory. Figure 5 *(up-right)* shows that a similar power relationship exists between target-based tangential velocity $v_T(t)$ and target-based density function $D_s(t)$ according to the *DPPLA* algorithm:

$$V_T(t) = K\left(\frac{1}{D_s(t)}\right)^{\gamma}$$
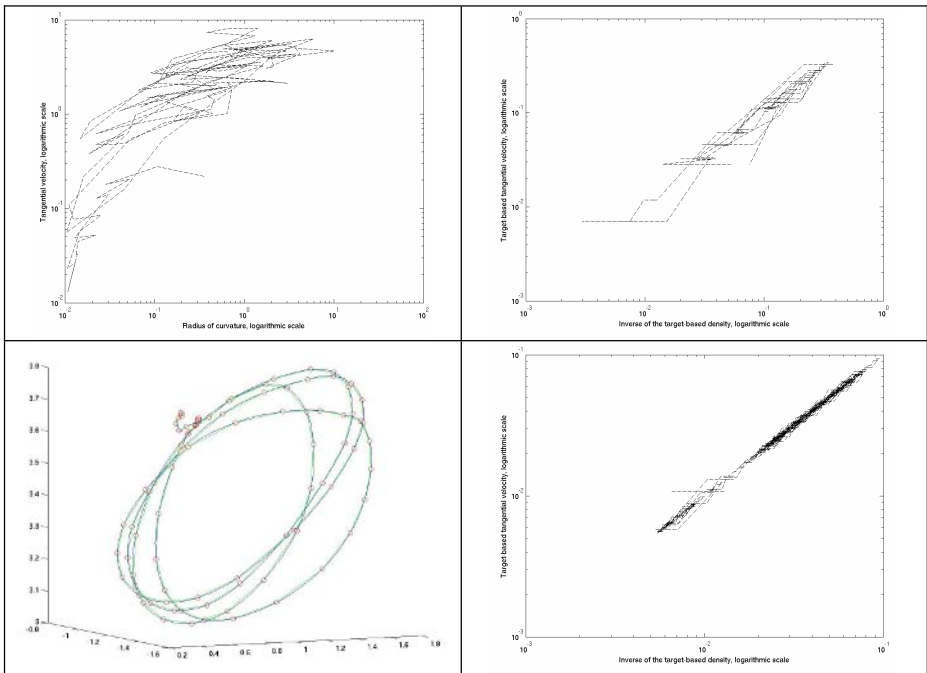
Fig. 5. *(up-left)* Tangential velocity versus radius of curvature in logarithmic scale; *(up-right)* Target-based tangential velocity versus inverse of target-based density in logarithmic scale. Both figures are traced for the complex pattern trajectory presented in Fig. 3; *(down-left)* End effector elliptical trajectory with target location; *(down-right)* Target-based tangential velocity versus inverse of target-based density in logarithmic scale.

with

$$V_T(t_i) = \frac{\|Tg_i - Tg_{i-1}\|}{t_i - t_{i-1}} = \frac{\delta Tg_i}{\delta t_i} \quad \text{et} \quad D_s(t_i) = \frac{1}{\delta Tg_i}$$

The power law between V_T and $1/D_s$ is also observed for an elliptical pattern trajectory (see Fig. 5 down-part) for which the γ coefficient has been estimated to be approximately 1.2. The degree of invariance of γ has not been established. In particular, it may depend on the type of performance (strength, speed, etc.).

6   Discussion

This study has dealt with an analysis-synthesis method that forwards the discretization of end-effector trajectories captured from human motion and provides the command input of a synthesis model controlling a hand-arm dynamical system. Our analysis method uses a Dynamic Programming Piecewise Linear Approximation (*DPPLA*) algorithm that extracts a sequence of multi-dimensional targets from the end-effector trajectory. These targets can be used as command input of a sensory motor dynamical system. The *DPPLA* algorithm automatically computes an optimal number of target points and their respective location along the motion frames.

This discretization algorithm at the command level can be defined to identify low-dimensional sub-sets of samples not uniformly distributed in motion time series. It can be used as a segmentation method to detect points where there is a higher complexity of the trajectory. It also provides a means of reducing the data flow at the command level. Finally, by discretizing the command input it becomes possible to concatenate successive elementary motion without having to deal with the transition mechanisms.

Experiments show the effectiveness of this discrete task-based analysis/synthesis approach. The analysis of motion capture data is conducted on a set of arbitrary end-effector trajectories. The compression rate can be fixed to a high level, while maintaining satisfactory simulation results. The discretized data are then used as command input of our sensory motor synthesis system. The animation is achieved on a hand-arm mechanical system with seven degrees of freedom, associated to a control policy based on a non-parametrical learning method. In the context of motion synthesis using a data-driven approach, it might be interesting to exploit discrete information that replaces motion capture data, without reducing the quality of the animation.

This discretization mechanism is not only a mathematical tool which aims to reduce the data flow at the task-level. Indeed, it can be pointed out that the adaptive sampling is correlated to invariant laws of movement. The *DPPLA* algorithm seems to exhibit a power relation between sampled tangential velocity and the inverse of the sampled density function. The properties of this relationship, in particular the degree of invariance of the power parameter need to be further explored. Nevertheless, preliminary results tend to highlight the pertinence of the discrete patterns

extracted from the *DPPLA* algorithm and in particular the link between the targets distribution along the trajectory and the curvature. In order to prove the generalization and the robustness of this law related to the *DPPLA* analysis method, more systematic tests should be conducted with various motion patterns performed by several subjects.

References

1. Vercher J.L.: Perception and synthesis of biologically plausible motion: from human physiology to virtual reality, Proceedings of Gesture Workshop 2005, (2005)
2. Kawato, M., Maeda Y., Uno, Y., Suzuki R.: Trajectory Formation of Arm Movement by Cascade Neural Network Model Based on Minimum Torque Criterion. Biological Cybernetics, Vol. 62. (1990) 275-288
3. Bullock, D., Grossberg, S., Guenther, F.H.: A Self-Organizing Neural Model of Motor Equivalent Reaching and Tool Use by a Multijoint Arm. Journal of Cognitive Neuroscience, Vol. 54. (1993) 408-435
4. Barbic J., Safonova A., Pan J.Y., Faloutsos C., Hodgins J., Pollard N.: Segmenting Motion Capture Data into Distinct Behaviors. In Proceedings of Graphics Interface 2004, (2004) 185–194
5. Boukir S., Chenevière F.: Compression and recognition of dance gestures using a deformable model, Pattern Analysis and Applications (PAA) Journal, Springer-Verlag, Vol. 7, No 3, (2004) 308-316
6. D. Chi, M. Costa, L. Zhao, and N. Badler: The EMOTE model for Effort and Shape, ACM SIGGRAPH '00, New Orleans, LA, (2000) 173-182
7. Gibet S., Marteau P.F.: A Self-Organized Model for the Control, Planning and Learning of Nonlinear Multi-Dimensional Systems Using a Sensory Feedback, Journal of Applied Intelligence, Vol. 4. (1994) 337-349
8. Lebourque T., Gibet S.: A complete system for the specification and the generation of sign language gestures. Lecture Notes in Artificial Intelligence, Lecture Notes in Artificial Intelligence 1739, in Gesture-Based Communication in Human-Computer Interaction, A. Braffort, R. Gherbi, S. Gibet, J. Richardson, D. Teil Eds., Springer-Verlag, Berlin Heidelberg (1999)
9. Gibet S., Lebourque T., Marteau P.F.: High level Specification and Animation of Communicative Gestures, Journal of Visual Languages and Computing, Vol. 12. (2001) 657-687
10. Julliard F., Gibet S., RML: A specialized Parallel Language for 3D Motion Control Specification., In International Applied Informatics Conference, Parallel and Distributed Processing Symposium, Innsbruck, Austria, (2001) 39-45
11. Gibet S., Marteau P.F., Julliard F.: Models with Biological Relevance to Control Anthropomorphic Limbs : A Survey. Lecture Notes in Artificial Intelligence. Lecture Notes in Artificial Intelligence 2298, Ipke Wachsmuth and Martin Fröhlich Eds., Springer Verlag, Berlin Heidelberg (2002) 105-119
12. Gibet S., Marteau P.F.: Expressive Gesture Animation Based on Non Parametric Learning of Sensory-Motor Models. CASA 2003, Computer Animation and Social Agents (2003)
13. Perez J.C., Vidal E.: Optimum polygonal approximation of digitized curves, *Pattern Recognition Letters*, Vol. 15. (1994) 743-750

14. Goodrich M.T.: Efficient piecewise-linear function approximation using the uniform metric. Proceedings of the tenth annual symposium on Computational geometry Stony Brook, New York, United States, (1994) 322 – 331
15. Agarwal P.K., Har-Peled S., Mustafa N.H., Wang Y.: Near-Linear Time Approximation Algorithms for Curve Simplification Proceedings of the 10th Annual European Symposium on Algorithms (2002)
16. Bellman R.: Dynamic Programming. Princeton University Press, Princeton, NJ (1957)
17. Lacquaniti, F., Terzuolo, C., Viviani, P.: The law relating the kinematic and figural aspects of drawing movements. Acta Psychologica, Vol. 54. (1983) 115-130

Simulation of Hemiplegic Subjects' Locomotion

Nicolas Fusco, Guillaume Nicolas, Franck Multon, and Armel Crétual

Laboratoire de Physiologie et de Biomécanique de l'Exercice Musculaire
Rennes 2 University, Av. Charles Tillon CS 24414, 35044 Rennes, France
{nicolas.fusco, guillaume.nicolas, franck.multon, armel.cretual}@uhb.fr

Abstract. This paper aims at describing a new method to simulate the locomotion of hemiplegic subjects. To this end, we propose to use inverse kinematics in order to make the feet follow a trajectory with respect to the root frame linked to the pelvis. The 11 degrees of freedom are then retrieved by inversing the kinematic function while taking other constraints into account. These constraints, termed secondary tasks impose that the solution ensures joints limits and energy minimisation. In addition to those general constraints, the main originality of this work is to take spasticity into account. This new constraint is obtained according to the specificity of the subject's pathology. The results show that angular trajectories for the pelvis, the hips and the knees for the simulated and the real motion are very similar. This preliminary work is promising and could be used to simulate the effects of reeducation or medical treatments on patients' gait.

Keywords: inverse kinematics, locomotion, hemiplegia, spasticity.

1   Introduction

Using computer simulation and animation in biomechanics is a new approach to performing motion analysis. Nevertheless, animation based only on kinematics could generate unrealistic movements. Hence, to simulate human locomotion, a large set of solutions is proposed in the literature [1]. Among these techniques, inverse kinematics is widely used to ensure foot-contact with the ground without sliding [2] or to adapt an existing movement to a different skeleton [3].

Several past works [4] have proposed an explicit method to solve inverse kinematics problems for limbs composed of only a few segments. These methods are efficient for isolated upper and lower limbs but seem difficult to apply to the whole body. To deal with more complex structures, the problem is generally solved using a linear approximation of the direct kinematics equation [5]. In some cases, several constraints can be applied, for example, controlling several extremities or constraining centre of mass movements. This problem can be solved by weighting each constraint [6]. [7] proposed a task-priority formulation to take all of these constraints into account.

Whatever the technique is, because of the redundancy, there generally exists an infinity of possible solutions. A secondary task is then proposed to select a specific solution. The main problem is to determine the constraints that will

help to compute a human-like movement. To this end, several methods have been proposed. [8] offered to select the solutions that are close to captured trajectories for arm movements. [9] added a weight matrix in the resolution process. The weights were identified by optimisation until the calculated movement resembled a captured one. All these approaches are based on real movements and cannot be used if no knowledge on the resulting movement is available (such as for simulating new behaviors). A more general approach is to control also the position of the centre of mass [10]. These methods are well adapted to quasi-static motions, but are not suitable for movements involving dynamics such as locomotion.

Simulating a human-like motion is a complex problem because a large number of parameters has to be considered, including energy, comfort, equilibrium, joint limits, muscle activation, etc. To simulate such specific locomotion of subjects with hemiplegia, the nature of the pathology also has to be considered. In this paper, we focus on hemiplegic subjects with spasticity. The spasticity engenders an increase in the stretching reflex linked to speed of muscle contraction. As a consequence, the locomotion of such subjects is generally dissymmetric with lower speeds than those of healthy subjects. [11] demonstrated that there is no correlation between dissymmetry and walking speed but that individual behaviors occurred. This result indicates that no general method can be directly applied to a patient. [12, 13] reported a lower hip flexion during the balance phase and a lower hip extension after foot-strike. They also pointed-out an exaggerated knee flexion at foot-strike and a lower knee flexion at the balance phase. The latter is explained by a deficiency of the motor controller and spasticity of the rectus femoris and the gastrocnemius. [14] demonstrated that cocontractions disorders influenced the metabolic energetic cost of walking in patients with cerebral palsy. Moreover, several authors demonstrated that hemiplegic patients generally spend more energy than healthy subjects do for walking at the same speed [15, 16]. However, this difference tends to decrease when walking speed increases [17]. In fact, the more the self-selected velocity of hemiplegic subjects is near that of healthy subjects, the more the difference tends to decrease. This means that self-selected speed is a reliable index of pathology level. Individual behaviors were again pointed-out. Of course, computer simulation makes it impossible to measure directly energy expenditure. However, [18] demonstrated that mechanical power estimates correctly the O_2 cost of walking in subjects with hemiplegia. As a consequence, energy requirements of a simulated walk could be approximated by the computation of the corresponding internal work.

In this paper, we propose a new approach to simulate locomotion of hemiplegic subjects with cerebral palsy. To do so, we propose to model spasticity through its effects on the stretching reflex that decrease joint limits and angular velocity. We focus on the rectus femoris and gastrocnemius that directly act on knee flexion and extension. The method described in sections 2 is applied to a skeleton that was fitted to anatomical landmarks taken on a real hemiplegic subject. This technique is based on inverse kinematics and secondary tasks are used to take pathology into account. In section 3, the simulated angular trajectories are then compared to those of this subject to evaluate the model's efficiency.

2   Modelisation

2.1   Inverse Kinematics to Model Human Locomotion

Given a set of anatomical landmarks measured on a real subject, we are able to construct the kinematic function that links the angular representation of the posture to the position of lower-body extremities. In this study, we focus on the lower-body including the pelvis, femurs and tibias. The root frame of the kinematic chain has its origin at the middle of the pelvis. Its orientation is fixed meaning it has the same orientation as the pseudo-Galilean frame linked to the laboratory. The pelvis can rotate along its main axes, i.e. it has 3 degrees of freedom (DOF). Each hip is associated to 3 new DOF which simulate the flexion/extension, adduction/abduction and medial and lateral rotation of this typical ball and socket joint. Each knee is considered as a hinge joint with 1 DOF corresponding to the flexion/extension. The system is thus composed of 11 DOF (Fig. 1). The root frame is in fact that of the pelvis at the initial posture with every angle set to zero.

Root frame

3 ROTATIONS

Pelvis frame

3 ROTATIONS   3 ROTATIONS

Left femur frame   *Right femur frame*

1 ROTATION   1 ROTATION

Left tibia frame   *Right tibia frame*
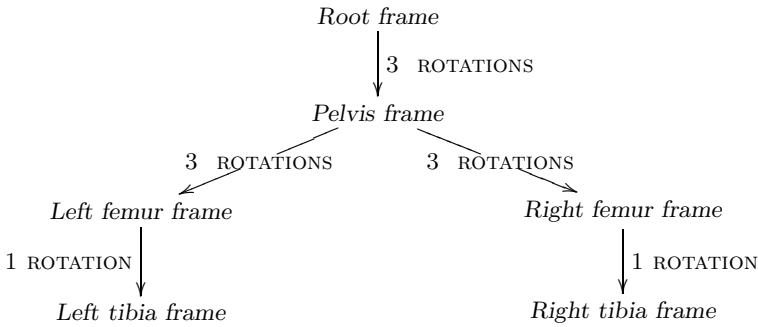
Fig. 1. Structure of the 11-DOF skeleton

Hence, the position of each ankle with respect to F is:

$$X = f(\theta) \tag{1}$$

where $X = (x_l, y_l, z_l, x_r, y_r, z_r)$ is the position of the effectors with respect to the Cartesian position of the left and the right ankle, θ stands for the 11-dimensional vector of angles applied to the DOF. The set of all θ is called configuration space. From equation 1, we numerically compute the Jacobian of the system:

$$\Delta X = J(\theta)\Delta\theta \tag{2}$$

Given, the trajectories of the ankles in the root frame, referred to as *poulaine* in the remainder of the paper, the primary task is ensured by inversing equation 2:

$$\Delta\theta = J^+(\theta)\Delta X \tag{3}$$

where J^+ is the pseudo-inverse of J. As 6 constraints are applied to this 11-DOF system, that leads to a 5-dimensional kernel space of J, termed *Ker*. This kernel is the image of the configuration space by the matrix $(I - J^+J)$, I being the unity matrix. Nevertheless, nothing ensures that the solution proposed with this equation verifies constraints such as joint limits or energy expenditure minimisation. For this, a secondary task z is generally proposed to take those constraints into account [5]:

$$\Delta\theta = J^+(\theta)\,\Delta X + \alpha\left(I - J^+J\right)\nabla z \qquad (4)$$

where α is a weight associated with the secondary task and z stands for a cost function to minimise. Generally, this secondary task is solved iteratively with the steepest descent method. On one hand, if several constraints are proposed concurrently, the system tries to solve a least square problem. A compromise of all the conditions is consequently found with this method. On the other hand, some of the constraints (such as preserving joint limits) require strict verification while others (such as energy expenditure) only need to be minimised.

2.2   Specifying Pathology Using the Secondary Task

When the primary task is solved we obtain an initial value $\Delta\theta_m$:

$$\Delta\theta_m = J^+\Delta X \qquad (5)$$

This solution has a minimal norm but nothing ensures that it respects joint limits and produces realistic trajectories. If $\Delta\theta_m$ is a solution of equation 3 and ϕ an element of *Ker*, then $\Delta\theta_m + \phi$ is also a solution of equation 3. The goal of the secondary tasks is to find the optimal ϕ that minimises a set of functions. Let us call θ_t the current value of θ. To model the specific gait of hemiplegic subjects, we chose the functions below:

- T1: To account for joint limits, we defined a continuous and derivable cost function that rapidly increases beyond the joint limits. In this paper, we propose an exponential function:

$$f_1\left(\theta_t, \Delta\theta_m, \delta\right) = \sum_{i=1}^{11}\left(e^{\zeta(\alpha_i - bup_i)} + e^{\zeta(blow_i - \alpha_i)}\right)$$
$$with \quad \alpha \quad = \left(\theta_t + \Delta\theta_m + (I - J^+J)\,\delta\right) \qquad (6)$$

  where bup_i and $blow_i$ are respectively the upper and the lower joint limits for the ith DOF. ζ is a constant coefficient that ensures a rapid increase when the angle is beyond the joint limits and a rapid decrease when it is within those limits (Fig. 2). δ is any element of the configuration space.

  For subjects with cerebral palsy causing knee flexion/extension disorders, the joint limits are customised. Hence, the maximum and minimum knee angle were obtained specifically for this subject. A new function taking the spasticity into account by also constraining the knee angular velocity with a maximum value has been defined. This value was obtained by computing
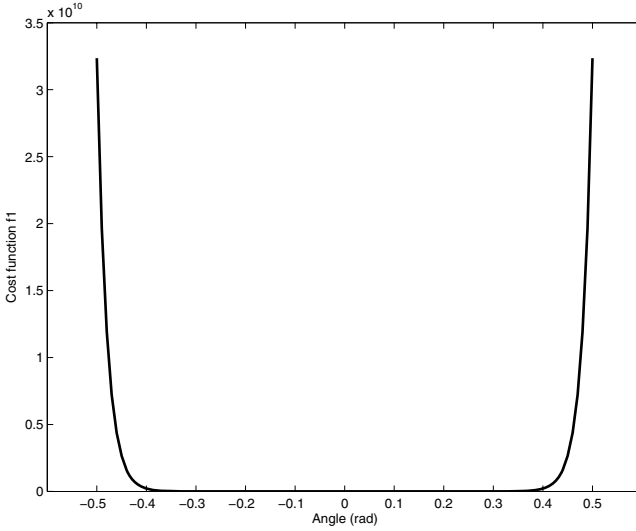
Fig. 2. Cost function f_1 depending on an angle giving a maximum and a minimum joint limit

the maximum angular velocity of the affected knee while walking at the maximum speed. This angular velocity was multiplied by an empirical factor 1.2 to take a 20 % offset into account. The resulting angular velocity can vary between 0 and max by adding a new function $f_{1_{bis}}$.

$$f_{1_{bis}}(\theta_t, \Delta\theta_m, \delta) = e^{\zeta(\dot{\alpha}_{knee} - vup)} + e^{\zeta(vlow - \dot{\alpha}_{knee})}$$

$$\text{with} \quad \alpha_{knee} = (\theta_t + \Delta\theta_m + (I - J^+ J)\delta)_{knee} \tag{7}$$

where α_{knee} is the knee angle of the affected side and $\dot{\alpha}_{knee}$ its angular velocity, vup and $vlow$ are respectively the upper and the lower knee angular velocity limits. ζ is a constant coefficient that ensures a rapid increase when the velocity is beyond the knee angular velocity limits and a rapid decrease when it is within those limits.

– T2: minimising the rotational kinetic energy of each body segment:

$$f_2(\theta_t, \Delta\theta_m, \delta) = \sum_{b=1}^{5} \left[\frac{1}{2} R_b I_b R_b^T \left(w_b \left((I - J^+ J)\delta, \Delta\theta_m \right) \right)^2 \right] \tag{8}$$

where b stands for the body segment index, I_b is the inertia of segment b. w_b is a function that computes the angular velocity vector of segment b depending on $\Delta\theta_m$ and the optimized parameter δ. R_b is the transform matrix between the body segment frame and the root frame, computed from θ_t.

– T3: searching for a solution close to the rest posture:

$$f_3\left(\theta_t, \Delta\theta_m, \delta\right) = \left\|\theta_t + \Delta\theta_m + \left(I - J^+J\right)\delta - \theta_r\right\|^2 \tag{9}$$

where θ_r is the angle at rest posture provided by motion capture on static trials.

We propose to use the Multidirectional Search (MDS) method [19] to solve this secondary tasks problem. It enables one to minimise cost functions wether there are derivable or not and is less sensible to local minimum than the steepest descent method.

According to an initial value which is the posture at the previous time step, MDS evaluates among a set of neighbours the cost function f_i. The neighbours are selected by using a simplex Δ that is linked to all the main axes of the search space:

$$\Delta_j = \{\forall i, \delta_j + \beta \cdot unit\,(i)\} \tag{10}$$

where δ_j is the current solution at step j, β stands for the size of the simplex and $unit(i)$ is a vector composed with zeros excepted for the ith element (equal to 1). Δ_j is a set of candidates that can be evaluated thanks to the cost function. In order to cover a wider space of research, two operators are used: a contraction ($\times$ 0.5 in our examples) and an expansion operator ($\times$ 2 in our examples). Those two operators modify the simplex size by scaling it by real factors respectively lower and greater than 1. Among all the resulting candidates, the system selects the one that minimises the cost function. This candidate becomes the new current solution δ_{j+1}. The process is repeated until a stable solution $\hat{\delta}$ is obtained or the cost function goes under a given threshold ϵ set to 10^{-4} in our example.

The optimal solution of the inverse kinematics problem is given at next step by:

$$\theta_{t+\Delta t} = \theta_t + \Delta\theta_m + \left(I - J^+J\right)\hat{\delta} \tag{11}$$

3   Experimental Validation

3.1   Experimental Set Up and Procedure

A volunteer with hemiplegia participated in this experiment after giving informed consent. The subject age, height and mass were 25 years, 1.8 m and mass 83 Kg respectively. This patient with cerebral palsy suffered from spasticity of the right rectus femoris and right gastrocnemius so that his locomotion was clearly affected. Three-dimensional kinematics of the subjects hemiplegic lower extremity were documented with the Vicon370 motion analysis system (product of Oxford Metrics, Oxford, UK). Seven infrared, 60 Hz cameras recorded the location of thirty reflective markers placed over standardized anatomical landmarks overlying the bony landmarks (Fig. 3). The subject's motion data was captured during gait test on a treadmill for walking speeds increasing from 0.5 to 1.1 m.s^{-1}. Finally, the *poulaines* which are the effectors' position for the
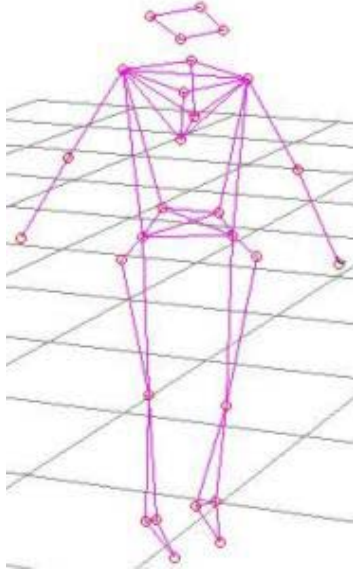
Fig. 3. Anatomical landmarks used for motion capture

primary task, were obtained on the subject walking at 0.5 m.s^{-1}. Given these *poulaines*, the problem is to calculate realistic angular trajectories.

In order to evaluate the resulting simulation, we proposed to calculate the Root Mean Square (RMS) which evaluates the difference between the simulated and the captured trajectories:

$$C_i = \sqrt{\frac{1}{T} \sum_{t=0}^{T} \left(\theta_i\left(t\right) - \hat{\theta}_i\left(t\right) \right)^2} \tag{12}$$

where t stands for the time and T for the total duration, i for the DOF and $\hat{\theta}_i$ for the captured trajectory. In this equation, $\hat{\theta}_i$ is shifted to θ_i mean value.

3.2 Results

First, Fig. 4 shows that the primary task is absolutely ensured. In fact, the RMS errors between the captured *poulaines* and those simulated are less than 1 mm.

Second, in Fig. 5, one can see simulated (continuous line) and experimental data (dashed line) angular trajectories for the hips and the knees. All the simulated trajectories have a similar shape compared to real ones.

Regarding the subject's pathological specificity, the simulated angular trajectories of the affected side show a lower hip flexion during the balance phase and a lower hip extension after foot-strike. However, the differences (around 0.1 rad)
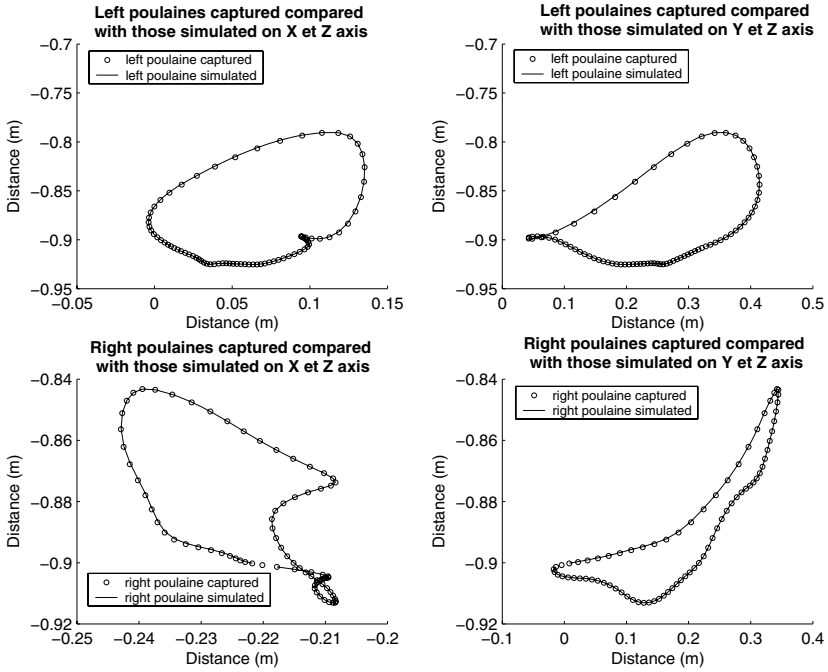
Fig. 4. Simulated and captured *poulaines* for the healthy and the affected leg, in the frontal plane on the left and in the sagittal plane on the right

with the healthy leg are small. Futhermore, as reported in previous works [20] the flexion of the healthy knee resembles a shape classically encountered with healthy subjects. However, the flexion of the affected knee is far from a classical shape encountered with healthy subjects. We observe an exaggerated knee flexion at foot-strike and a lower knee flexion at the balance phase on the affected side compared to the healthy one. The maximal flexion of the healthy side is 1.15 rad against 0.74 rad for the affected side.

Third, for the differents joints, as those reported in Table 1, the results showed are very close to the real trajectory. Indeed, the RMS error value between the simulated and the real angular trajectory is lower or equal to 0.11 rad.

In the same way, we compare the RMS error between the simulated Cartesian position of each joint with those of the real movement. Again, Table 2 reported error equal or less than 10 mm. These results indicate that Cartesian positions are ensured by the simulation.

The results exhibit very similar Cartesian position while the angular trajectoriese simulated are very close to real ones for all the articulations. Nevertheless, some differences occur. This seems to be mainly due to two reasons. First, the application of the captured, and thus, noisy *poulaine* to the virtual patient. This *poulaine* is obtained thanks to markers that slide over the skeleton. As proof of this sliding, for some points, we obtain an instantaneous distance between
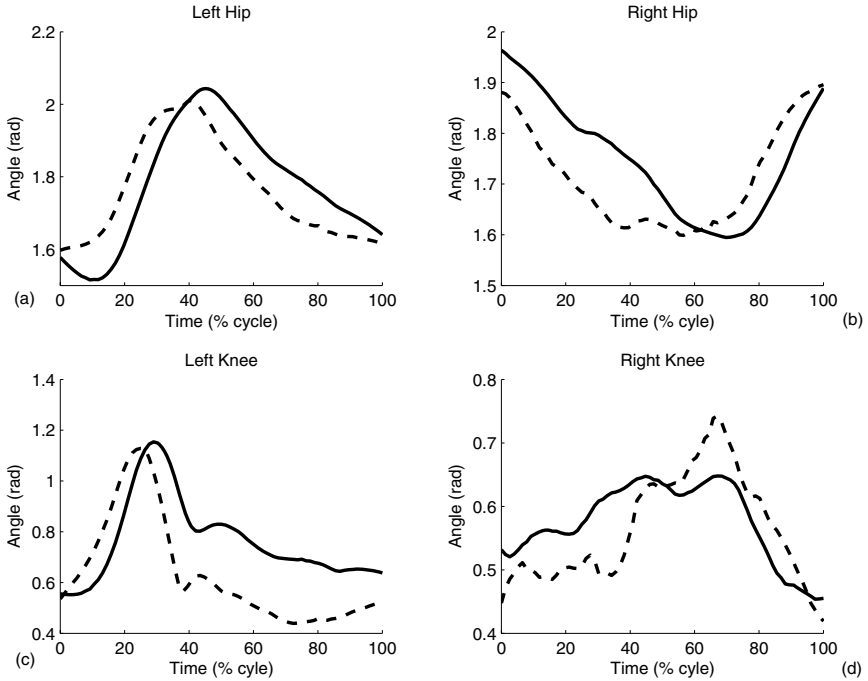
Fig. 5. Simulated (continuous line) *vs.* captured (dashed line) angular trajectories for the hips (a and b) and the knees (c and d)

Table 1. RMS error between the simulated and the real angular trajectories

Joints' angles	RMS (rad)
Pelvis inclination	0.03
Pelvis obliquity	0.10
Pelvis int/ext rotation	< 0.01
Left Hip fle/ext	0.01
Left Hip abd/add	< 0.01
Left Hip int/ext rotation	0.03
Right Hip fle/ext	0.01
Right Hip abd/add	< 0.01
Right Hip int/ext rotation	0.10
Left Knee fle/ext	0.11
Right Knee fle/ext	< 0.01

markers greater than dimensions calculated in the static trial. For example, the minimum knee flexion is equal to 0.45 rad for the healthy knee in the real angular trajectories. This is a very high value since this time corresponds to a foot-strike where the knee is supposed to be quite extended. This problem could be solved

Table 2. RMS error between the simulated and the real postion trajectories

Joints' position	Axis	RMS (mm)
Left trochanter	x	4
	y	9
	z	5
Right trochanter	x	3
	y	5
	z	< 1
Left Knee	x	1
	y	10
	z	4
Right Knee	x	1
	y	1
	z	< 1

by retrieving actual joint centres instead of directly using external markers. Second, we may implicate the choice of the secondary task. In fact, the secondary task does not take into account all the parameters of the subject's gait and his pathology.

4   Discussion

We described a preliminary work to simulate the gait of hemiplegic subjects with cerebral palsy. The pathology of a subject was modeled through constraint equations. Those equations are obtained according to the subject's pathology specificity. The subject had a spasticity of the rectus femoris and of the gastrocnemius that engendered knee flexion disorders. We consequently proposed to model its pathology by adding a function limiting his minimum and maximum knee flexion and knee flexion velocity.

This methodological work obviously requires validation on a wider set of subjects. Moreover, the *poulaine* that is applied as an entry of our model also contains information linked to the subject pathology. To validate our model, it could be interesting to use other *poulaines*: ranging from other subjects with cerebral palsy to healthy one.

The possible applications of such a model are linked to reeducation.

First, it could allow us to carry-out fundamental research on the links between all disorders encountered with hemiplegic subjects. Hence, in the literature, the main problem is to define cause-to-effect links between observable phenomena [18]. Nevertheless, as locomotion is the consequence of a large set of coupled parameters, it is impossible to isolate the effect of one of them on another. With simulation, it is possible to change only one parameter and verify its consequences. For example, are the knee flexion disorders greater than step duration dissymmetry for energy expenditure?

Second, for reeducation, this kind of model could be used to evaluate possible consequences of several methods on energy expenditure. In the literature, it

is demonstrated that knee flexion disorders causes a raising of the trunk and the pelvis in the frontal plane during the balance phase of the affected leg. To overcome this problem, it has been proposed to use controlateral shoe-lifts that allow the subject to balance the leg without touching the ground and rotating the pelvis. If the goal is to decrease energy expenditure while walking in everyday life, is this strategy more interesting than retrieving a larger knee flexion?

For both applications, the *poulaine* could be captured on the patients so that the main limitation of our method is lowered. Indeed, the goal is to have a computer model walk as the real model while only changing a minimum set of parameters (increasing joint limits, scaling the *poulaine* to increase the step length, time-scaling the *poulaine* to obtain symmetrical cyles for examples).

Acknowledgements

References

1. Multon, F., France, L., Cani-Gascuel, M., Debunne, G.: Computer animation of human walking: a survey. Journal of Visualization and Computer Animation **10** (1999) 39–54
2. Boulic, R., Thalmann, D.: Combined direct and inverse kinematic control for articulated figures motion editing. Computer Graphics Forum **11** (1992) 189–202
3. Monzani, J., Baerlocher, P., Boulic, R., Thalmann, D.: Using an intermediate skeleton and inverse kinematic for motion retargeting. In: Eurographics, Interlaken (2000)
4. Tolani, D., Badler, N.: Real-time inverse kinematics of the human arm. Presence, Teleoperators, and Virtual Environments **5** (1996) 393–401
5. Baerlocher, P.: Inverse kinematics techniques for the interactive posture control of articulated figures. PhD thesis, EPFL, Switzerland (2001)
6. Philips, C., Zaho, J., Badler, N.: Interactive real-time articulated figure manipulation using multiple kinematic constraints. Computer Graphics **24** (1990) 245–250
7. Baerlocher, P., Boulic, R.: Task-priority formulations for the kinematic control of highly redundant structures. In: IEEE IROS'98. (1998) 323–329
8. Wang, X., Verriest, J.: A geometric algorithm to predict the arm reach posture for computer-aided ergonomic evaluation. The Journal of Visualization and Computer Animation **9** (1998) 33–47
9. Zhang, X., Kuo, A., Chaffin, D.: Optimization-based differential kinematic modeling exhibits a velocity-control : strategy for dynamic posture determination in seated reaching movements. Journal of Biomechanics **31** (1998) 1035–1042
10. Boulic, R., Mas, R., Thalmann, D.: Complex character positioning based on a compatible flow model of multple supports. IEEE Transactions on Visualization and Computer Graphics **3** (1997) 241–261
11. Feys, H., De Weerdt, W., Nieuwboer, A., Nuyens, G., Hanston, L.: Analysis of temporal gait characteristics and speed walking in stroke patients and control group. Musculoskeletal Management **1** (1995) 73–85

12. Olney, S., Griffin, M., Monga, T., Mc Bride, I.: Work and power in gait of stroke patients. Archives physical medicine and rehabilitation **72** (1991) 309–314
13. Olney, S., Richards, C.: Hemiparetic gait following stroke. part i: Characteristics. gait and posture **4** (1996) 136–148
14. Viswanath, B., Dowling, J., Frost, G., Bar-Or, O.: Role of cocontraction in the o2 cost of walking in children with cerebral palsy. Medicine and Science in Sports and Exercise (1996) 1498–1504
15. Olney, S., Monga, T., Costigan, P.: Mechanical energy of walking of stroke patients. Archives physical medicine and rehabilitation **67** (1986) 92–98
16. Waters, R., Mulroy, S.: The energy expenditure of normal and pathologic gait. Gait and Posture **9** (1999) 207–231
17. Zamparo, P., Francescato, M., De Luca, G., Lovati, L., Di Prampero, P.: The energy cost of level walking in patients with hemiplegia. Scandinavia Journal of Medicine and Science in Sports **5** (1995) 348–352
18. Viswanath, B., Dowling, J., Frost, G., Bar-Or, O.: Role of mechanical power estimates in the o2 cost of walking in children with cerebral palsy. Medecine and Science in Sports and Exercise (1999) 1703–1708
19. Torczon, V.: Multi-directional Search: A Direct Search Algorithm for Parallel Machines. Ph.d. thesis, Rice University, Houston,Texas, USA (1989)
20. Burdett, R., Skrinar, G., Simon, S.: Comparison of mechanical work and metabolic energy consumption during noram gait. Journal of Orthopaedic Research **1** (1983) 63–72

Handiposte: Ergonomic Evaluation of the Adaptation of Physically Disabled People's Workplaces

Frédéric Julliard

Centre Européen de Réalité Virtuelle,
25 rue Claude Chappe, 29280 Plouzané, France
frederic.julliard@enib.fr
http://www.enib.fr/~julliard

Abstract. This paper presents a virtual reality application dedicated to the ergonomic evaluation and adaptation of workplaces destined for the physically disabled. Handiposte aims at assisting doctors and ergonomists as an interactive simulation based tool. After a general survey about virtual reality tools oriented towards ergonomic studies, we propose a specific framework for the design of such an application using a particular design methodology. We conclude by presenting a first prototype and by outlining future improvements.

1 Introduction

In recent years, an increasing interest in promoting the participation of people with disabilities in working life has been shown. For example, ergonomic studies could help design equipment and work arrangements to improve working posture and ease the load on the body of a disabled worker; thus reducing instances of repetitive strain injury work related upper limb disorders. Ergonomists often perform their analyses manually: on the one hand, it consists of measuring a sample of workplace occupational features such as distances or angles and comparing them with worker's anthropometrical data (physical analysis). On the other hand, dynamical or procedural characteristics can be considered such as fatigability level of a task due to its repetition level and its energy requirement (physiological analysis). The evaluation process is then performed from a set of ergonomics rules to improve physical and physiological 'fit' between disabled people and the equipement they use. Handiposte aims at assisting doctors and ergonomists as an interactive simulation based tool. It allows to evaluate both the physical and the physiological fitness between the disability level of a worker and its workplace.

2 Previous Works

Only few works have focused on using immersive and interactive virtual reality tools to facilitate the evaluation of people's workplaces. Badler [1] has proposed a

computer aided ergonomic tool called E-Factory to support the design of industrial workplaces. It provides capabilities to detect collisions between the human and the environment and to analyse reachability to ensure feasibility of human tasks. A large number of humanoids and various direct or inverse kinematics models are provided, but the objectives are quiet more different from ours: task optimization and healthy criterions are firstly considered for able bodied people. The tool is not also well suited for computer neophytes because of the designing process which requires specific abilities in computer aided design.

3   Approach

As ergonomics deals with the interaction of technological and work situations with the human being, Handiposte relies on a specific approach which dissociates the workplace definition from the characteristics of the physically disabled person (Figure 1).
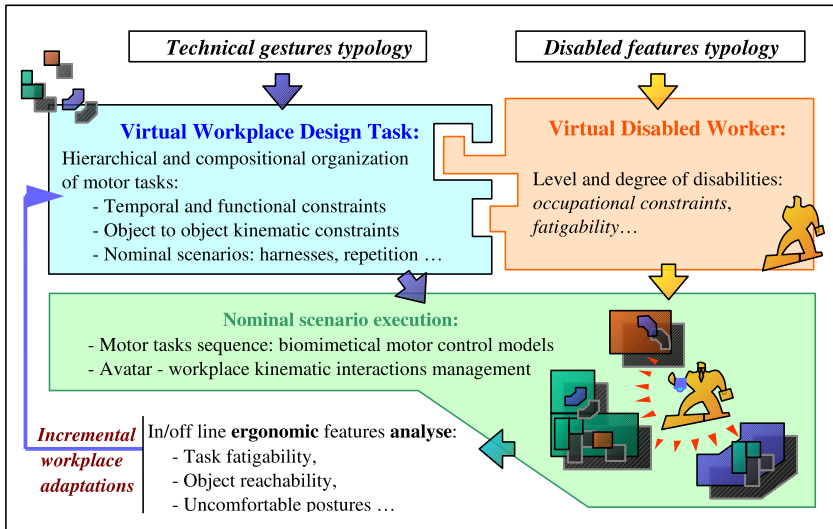
Fig. 1. A twofold typology based approach

On the one hand, the workplace definition is reduced as a gesture typology which expresses how occupational, functional and procedural features are combined. That is to say that the working environment is described from a compositional and hierarchical set of elementary objects. To handle virtual worker interations, each of these objects are described from their own degrees of freedom and are controlled by specific finite state machines.

On the other hand, level and type of the disability are firstly anthropomorphically expressed as a geometrical H-ANIM model where admissible ranges of

angles, lengths of segments or absence of limb are specified [7]. Other physiological features such as recovery time fill out this disability description.

Then, the interactive and participartory simulation consists of performing the calculation of the fitness function between workplace characteristics and the disability level. To achieve this goal, the disabled humanoid runs a nominal scenario viewed as a sequence of artificial gestures.

The previous approach significantly reduces the workplace design process by focusing on ergonomic characteristics, and not on geometrical features.

4   Application

A first prototype based on the virtual reality C++ library ARéVI [6] has been proposed in compliance with the previous methodology. It implements the following functionality concerning both workplaces and motor control representations:

– An XML based language supports the design of the working environment. Kinematic constraints and objects behaviors are thus hierarchically combined in order to determine the way objects react to worker actions.
– An inline analyzing module allows to detect unreachable and uncomfortable postures. It also permits to display physiological measures such as kinetic energy expenditure.
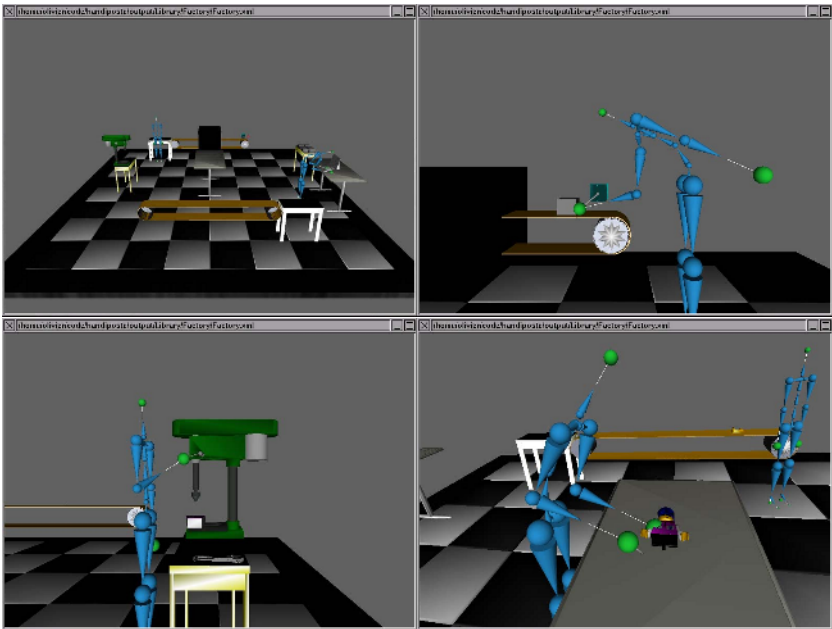

Fig. 2. A first prototype implementing a lego-man assembling process

– A set of motor control models based on direct and inverse kinematics approaches are used. These techniques have notably to be improved toward a biomimetical approach with respect to natural motor control laws in order to improve the accuracy of the analysing process [2] [3]. Theses motor control systems have also to be assisted with a fatigue evaluation module such as the one proposed in [4] where fatigue is derived from three main factors: task repetitiveness, comfort level [5] and effort depending on energy expenditure.

Figure 2 shows an assembly factory implemented with HandiPoste. In this example, a lego-man has to be jointed from several parts.

5 Conclusion and Future Works

The originality of the approach relies on a specific twofold typology. It allows for the computer neophyte to design workplaces in a hierarchical and compositional manner by associating a functional semantics to each object in term of technical gestures. The semantics of the current gesture typology has notably to be refined and generalized in order to be applicable to a larger number of workplaces.

The second remaining part of this work concerns the design of artificial gesture models whose main objectives consists of expressing in a biomimetically relevant view ergonomic factors such as performance, fatigability, hardness and compensation strategies through specialized motor control models.

References

1. Badler, N.: LiveActor: A virtual training environment with reactive embodied agents, Workshop on Intelligent Human Augmentation and Virtual Environments, Univerity of North Carolina at Chapel Hill, October 2002.
2. Gibet, S., Marteau, P.F., Julliard F.: Models with Biological Relevance to Control Anthropomorphic Limbs: a survey, In proceedings of International Gesture Workshop 2001, 105–119, 2002.
3. Gibet, S., Marteau P.F.: A self-organized model for the control, planning and learning of non-linear multi-dimensional system using a sensory feedback, Journal of Applied Intelligence, 4, 337–348, 1994.
4. Rodriguez, I., Boulic, R., Meziat, D.: A Joint Level Model of Fatigue for the Postural Control of Virtual Humans, Journal of 3D Forum, 1, 17, 70–75, March 2003.
5. Yang, F., Ding, L., Yang, C., Yuan, X.: An algorithm for simulating human arm movement considering the comfort level, Simulation Modeling Practice and Theory, 2005.
6. Harrouet, F.: ARéVI: Atelier de Réalité Virtuelle, http://www.enib.fr/~harrouet
7. H-ANIM: Humanoid Animation Working Group, http://www.h-anim.org

Modeling Gaze Behavior for a 3D ECA
in a Dialogue Situation

Gaspard Breton, Danielle Pelé, Christophe Garcia,
Franck Panaget, and Philippe Bretier

France Telecom Research and Development,
IRIS Laboratory,
4 rue du Clos Courtel,
35512 Cesson Sévigné Cédex, France
FirstName.LastName@francetelecom.com
http://www.francetelecom.com

Abstract. This paper presents an approach to model the gaze behavior of an Embodied Conversational Agent in a real time multimodal dialogue interaction with a group of users. The ECA's gaze control results from the merge of the outputs of a rational dialogue engine based on natural language interaction and face tracking of users.

1 Introduction

Social behavior is very important in face-to-face communication and is conveyed, for a large part, through gaze. In our current research, we are interested in modeling gaze behavior in order to increase face to face communication between a group of humans and a 3D Embodied Conversational Agent. It has been show [1] that gaze patterns can be extracted and correlated to dialogue structure in terms of turn-taking and nature of propositions (theme/rheme).

This work, based on [2], is an attempt to make a complete system taking into account discourse structure, multi-users face tracking and eyes/neck coordination. The system is made of a behavior engine that takes input from a dialogue engine and a face tracking system. The behavior engine receives locations of the users face from the face tracking system. It also receives information about dialogue structure from the dialogue engine. Then, it computes the actions to be performed and send the command to an animation system.

In the following, we mainly describe the behavior engine. We describe the face tracking system and the dialogue engine. We mainly focus on the strategy used by the behavior engine to gaze at the users.

2 Behavior Engine

The behavior engine is made of parallel hierarchical automates. It controls the ECA at several levels from biological needs, such as eyes blinking, to higher levels such as

expressions display. In this paper we only explain the hierarchy of automates that is used for gaze modelling.

At the higher level, the gaze controller selects the action to be performed among the followings **GazeAtUsers**, **GazeAway**, **GazeAnywhere** based on the studies from [3, 4]. The gaze controller is made of five composite states **Listening**, **Idle**, **Talking**, **Rheme** and **Theme**, each of them is composed of three sub-states **Start**, **Run** and **End**, making a total of 15 states. In each state, different intervals of probability are assigned to actions and a random drawing is performed to select the final action to be executed. This algorithm is fully described in another paper to be published at IUI2006.
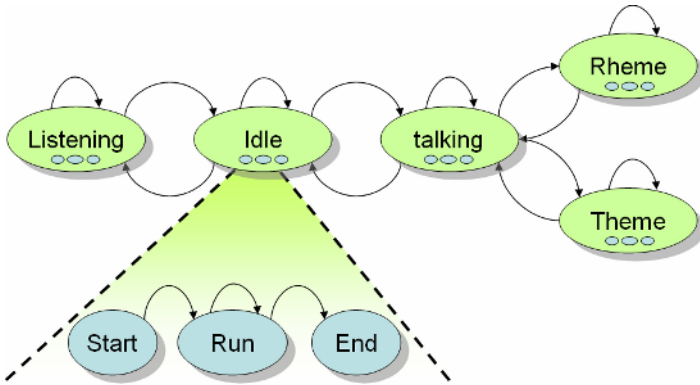
Fig. 1. Gaze controller states
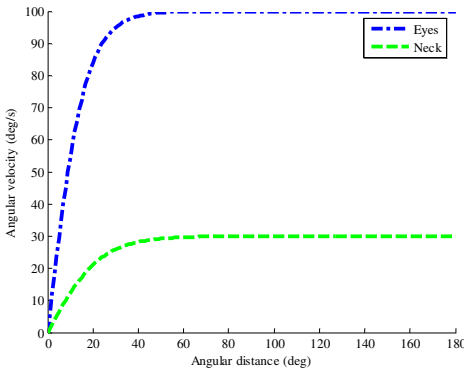
Fig. 2. Angular velocities curves
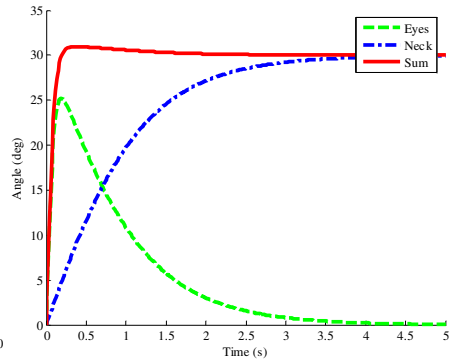
Fig. 3. Vestibulo-ocular reflex achieved by our system for a 30° target angle

The second level, gaze servoing is used to control the movements over time. It also performs the task of selecting targets (for example selecting an imaginary target for **GazeAnywhere**). Finally, it makes sure that each user is being gazed at one after the other, taking into account users' distance from the avatar, and already performed gazing times when performing the **GazeAtUsers** action.

At the lower levels, the neck and eyes automates compute the parameters to be sent to the animation engine. The eye automate has the neck automate as parent, so eye control is considered as a subtask of neck control. This is because the main goal is to move the head in front of the target. The eyes, which move faster, adapt to the neck movements to track the best the users. In order to reproduce the vestibulo-ocular reflex described by [5], the neck and eyes angular speeds are computed at each time step as a function of distance (see Fig. 2). The result is that the eyes immediately focus on the target whereas the head, which is heavier, point at the target later. As the head moves toward the target, the eyes come back to the rest position as can be seen on Fig. 3.

3   Dialogue Engine

In order to drive the behavior engine, the dialogue engine must be able to insert additional information into the output utterances. At least, the dialogue engine should be able:

- At a first level, to manage the global behavior of the avatar ;
- At a second level, to mark at least the thematic/rhematic structure.

In our implementation, the natural dialogue engine *Artimis* was used. This latter provides a generic framework to instantiate intelligent dialogue agents [6]. *Artimis* produces xml compliant output where theme and rheme clauses are clearly identified with <THEME> ... </THEME> <RHEME> ... </RHEME> tags.

4   Face Tracking

The incoming webcam video stream is analysed using the Convolutional Face Finder system (CFF) described in [7], which is able to robustly detect, in real time, multiple highly variable face patterns, of minimal size 30x30 pixels, rotated up to ±20 degrees in image plane and turned up to ±60 degrees. The CFF system returns the bounding boxes enclosing each detected faces. According to an average face model and calibration parameters estimated for the webcam, 3D coordinates of the detected face centers are estimated in the avatar coordinate system. Faces are tracked over time in a sequence of successive frames, taking into account possible detection misses and faces entering and exiting from the webcam field of view.

5   Conclusion and Future Work

In this system, we worked toward the integration of several aspects of gaze management from the highest, target selection, to the lowest levels, vestibule-ocular reflex. The algorithm was also designed in order to gaze at several users in case information need to be delivered to a group of people.

An evaluation has been conducted with users from of our research team and the system has been considered as producing convincing behaviors. We are starting in depth tests with a group of independent subjects. We are also planning to add more input to the system, such as, if the user is talking or not and also users positions from a 3D audio tracking system. It is also planned to enhance the animation by controlling pupils dilatation and eyelids movements.

References

[1] O. E. Torres, J. Cassel, and S. Prevost, "Modeling Gaze Behavior as a Funtion of Discourse Structure," presented at Workshop on Human-Computer Conversations, Bellagio, Italy, 1997.

[2] N. Courty, G. Breton, and D. Pelé, "Embodied in a look: bridging the gap between humans and avatars," presented at Proceedings of IVA, Irsee, Germany, 2003.

[3] J. Cassel, Y. I. Nakano, T. W. Bickmore, C. L. Sidner, and C. Rich, "Non-Verbal Cues for Discourse Structure," presented at Proceedings of the 41st Annual Meeting of the Association of Computational Linguistics, Toulouse, France, 2001.

[4] C. Pelachaud, V. Carofiglio, B. De Carolis, and F. De Rosis, "Embodied contextual agent in information delivering," presented at AAMAS, Bologna, Italy, 2002.

[5] D. Robinson, "The mechanics of human saccadic eye movements," *Journal of Physiology*, vol. 174, pp. 245-264, 1964.

[6] D. Sadek, P. Bretier, and F. Panaget, "ARTIMIS: Natural dialogue meets rational agency," presented at Proceedings of the 15th International Joint Conference on Artificial Intelligence, Nagoya, Japan, 1997.

[7] C. Garcia and M. Delakis, "Convolutional Face Finder: a Neural Architecture for Fast and Robust Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 1408-1422, 2004.

Playing "Air Instruments": Mimicry of Sound-Producing Gestures by Novices and Experts

Rolf Inge Godøy, Egil Haga, and Alexander Refsum Jensenius

University of Oslo, Department of Musicology,
P.O. 1017 Blindern, N-0315 Oslo, Norway
{r.i.godoy, egil.haga, a.r.jensenius}@imv.uio.no

Abstract. Both musicians and non-musicians can often be seen making sound-producing gestures in the air without touching any real instruments. Such "air playing" can be regarded as an expression of how people perceive and imagine music, and studying the relationships between these gestures and sound might contribute to our knowledge of how gestures help structure our experience of music.

1 Introduction

With the exception of "classical music" contexts, where it is generally considered taboo for listeners to make movements during public performances, listeners often spontaneously move their bodies, e.g. dance, tap their feet, nod their heads, make gestures with fingers, hands, and arms, etc. One category of such movements is known as playing "air instruments", e.g. "air guitar", "air drums", and "air piano", meaning making sound-producing gestures without making physical contact with any instrument, hence playing "in the air". Often done in private or semi-private settings (e.g. a pianist "playing" through a piece of music when trying to recall it, or someone making an air drum performance to the music at a party), some people also take the performance of air instruments very seriously. This is apparent in national and international air guitar championships, where the mimicry of sound-producing gestures (as well as other movements and expressions) is developed to high levels of sophistication.

Besides demonstrating strong personal involvement with the music, we believe air instrument playing shows some important principles of the mental coding of musical sound for non-musicians (novices) and musicians (experts) alike. We believe that images of sound-producing gestures are an integral part of the perception of musical sound, i.e. of identifying, discriminating, grouping, or doing "auditory scene analysis" [1] of musical sound, as well as of remembering, recalling and imagining musical sound, i.e. of musical imagery [2]. In taking air playing seriously, we assume that what can be observed of *overt behavior*, also reflects some essential features of *covert mental images* associated with musical experience.

When observing people playing air instruments, distinctions between sound-producing gestures and other kinds of gestures may not always be so clear-cut. Initially, we define *sound-producing gestures* as human movements made with the intention of transferring energy from the body to an instrument, i.e. as *excitatory gestures*, as well as human movements made with the intention of modifying the resonant features of an instrument, i.e. as *modulatory gestures* [3]. We have excitatory gestures such as hitting, stroking, bowing, blowing, kicking, etc., and modulatory gestures such as shaking, flexing, deforming or moving a mute. Furthermore, these gestures can have various *modes of execution*, such as fast, slow, hard, soft, short, long, etc., evident in several music-related metaphors (e.g. "hammering", "sweeping", "caressing"). These various modes of execution are often associated with what we like to call *amodal, affective or emotive gestures*, which may potentially include all the movements and/or mental images of movements associated with more global sensations of the music, such as images of effort, velocity, impatience, unrest, calm, anger, etc. In observing air instrument playing, such amodal, affective or emotive gestures often tend to fuse with sound-producing gestures in the more strict sense (i.e. excitatory and/or modulatory gestures). In some cases of air playing we may also see more vague *sound-tracing gestures*, such as in following melodic contours, rhythmical/textural patterns or timbral/dynamical evolutions with hands, arms, torso, or whole body. Such gestures could be understood as reflecting the total sonic evolution of the music more than the assumed sound-producing gestures (see [4] for a more extensive discussion of gesture categories).

Air playing gestures may often be quite approximate or sketch-like, posing several theoretical and methodological challenges (see sections 4 and 5 below), but this vague, and inexact nature of air playing is also what we find so intriguing. Observing how even novices make spontaneous air playing gestures which largely match the music, makes us believe that there are important links between musical sound and gestures in need of serious study. In the following sections we will present some theoretical considerations, an account of observation studies we have carried out, and some remarks on how we understand air playing in the context of music cognition.

2   Auditory-Gesture Links

For trained musicians, the link between sounds and sound-producing gestures are in most cases immediate and even involuntary [5]. Most musicians will probably agree that making, or merely imagining, sound-producing gestures is an efficient strategy for recalling music, or even planning and carrying out musical improvisation [6]. From such practical accounts, as well as from some experimental evidence [7], it seems reasonable to claim that musical memory includes procedural memory, i.e. memory for gestures, as well as auditory memory, i.e. memory for sound. However, we believe there are more general reasons for the close auditory-gesture links that we are studying here.

From an "ecological" perspective, it seems quite clear that auditory perception makes use of a number of cues and experience-based schemata when trying to make sense of sound. In particular, identification of sound source, what Bregman calls *stream segregation* [1], is important for making sense of the complex mass of sounds that we are exposed to. Sounds are associated with causality, hence with both sound-producing actions and resonating objects. As for resonating objects, such as strings, tubes, plates, membranes, etc., we seem to posses a considerable amount of "everyday" knowledge of features associated with various materials and shapes, e.g. "metallic", "soft", "hard", "hollow", etc. Likewise, we seem to have extensive ecological knowledge of the excitatory and modulatory gestures used to generate sounds [8].

One of the most significant efforts to explore auditory-gesture links can be found in the so-called "motor theory" of perception in linguistics [9, 10]. This theory has claimed that language perception, as well as language acquisition, is based on learning the articulatory gestures of the human vocal apparatus. In other words: we can make sense out of what we hear because we guess how the sounds are produced. Although this motor theory has been controversial, recent neuro-imaging studies seem to support the idea of perception as an active process involving motor cognition [11, 12]. There have also been suggestions of close evolutionary links between speech sounds and gestures [13], and research on gestures in speech contexts suggests that gestures not only are supplementary to the verbal content, i.e. an element for added expression and emphasis [14], but also instrumental in facilitating or even generating speech [15]. Lastly, we believe ideas from recent neuro-cognitive research on motor elements in perception and cognition in general [16], fit quite well with the idea that there are close links between sound and gestures. This neuro-cognitive research suggests that we regard perception and cognition as an incessant simulation and re-enactment of our impressions of the external world and of our bodies, implying that a mental "re-play" of sound-producing gestures would be part of making sense of sound.

3   Motormimetic Sketching

Combining the term *motormimetic*, denoting the imitation of "real" sound-producing gestures, and *sketching*, indicating the approximate nature of the imitation, we end up with the expression *motormimetic sketching*. Motormimetic sketching can be an activity of both novices and experts, generating quite approximate, yet in our opinion, significant images of musical objects.

Imitating what we believe others are doing, either overtly or covertly, is increasingly regarded as fundamental not only to learning and socialization, but also for understanding what others are doing [17, 18]. Covert imitation is understood to be at work whenever we see and/or hear others acting (although, in some cases, children, as well as people with some mental disorders, may exhibit overt imitation). Imitation, understood as a persistent activity when perceiving the actions of others, seems to go quite well with the abovementioned motor

theory of perception, i.e. that we mentally simulate the actions of others when we are trying to make sense of the sounds they make.

Air instrument playing, understood as motormimetic sketching, is then an egocentric, "I do" type of activity, imitating assumed sound-producing gestures of even quite complex musical objects, and also by people who would in no way be able to reproduce the heard music on an instrument. Thus, we speak of a *novice to expert continuum* in this motormimetic sketching, as opposed to a more sharp distinction we would make between people unable, and people able, to play "real" instruments. One objective of our studies is to explore these approximate renderings of sound-producing gestures by novices, as we believe this could teach us something about how people who do not have any musical training (and who even regard themselves as "unmusical") perceive significant global features in the music they hear.

As for the phenomenon of sketching, we were surprised to find so little research within the cognitive sciences that dealt with this subject. The most relevant discussions of sketching we have found are either in more art-oriented [19] or in design-oriented literature[20]. As we know from sketching in the visual arts, we may find a sketch quite salient, and well representing what it is supposed to depict, in spite of the rather sparse number of pencil strokes. We may thus speak of sketching, in the context of gestures, as a kind of "goal-directed imitation", what is called GOADI in [21], meaning that people (both children and adults) seems to initially focus on some goal-points when imitating gestures.

In our context, we understand the phenomenon of motormimetic sketching as follows: On first listening, we can make a spontaneous and quick tracing of assumed sound-producing gestures, reflecting the rough outline and global feeling (mood, sense of effort, sense of speed, etc.) of the music. Subsequent listening will help in gradually refining and adding detail, but the overall shape and character is usually manifest in the course of the first listening. In this way, motormimetic sketching is a kind of top-down activity, as the overall shapes of the gestures may set the frames for progressively finer details in the air playing.

4   Observation Studies of Air Piano Playing

To find out more about motormimetic sketching as a phenomenon, as well as some associated theoretical and methodological issues, we conducted a series of observation studies of air piano playing.

Subjects and sessions. Five persons with different musical and movement-related training were recruited for the observation studies:

A. Novice. No musical or movement-related training.
B. Intermediate. Some musical training on different instruments, and some movement-related training.
C. Semi-expert. Extensive musical training on several instruments and university level music studies, but no movement-related training.
D. Semi-expert. Extensive musical training on piano and university level music studies, but no movement-related training.

E. **Expert.** Professional pianist with extensive university level training in performance, but no movement-related training.

All subjects were informed about the purpose of the study, as well as how the sessions were going to be conducted. This included explicit instructions about trying as best they could to play air piano, by focusing their attention towards making what they believed to be the sound-producing gestures best fitted to the music they were going to hear. They were also told that the musical excerpts might or might not be familiar to them, and that their initial air playing gestures probably would come *after* the corresponding sounds, but as each excerpt would be played three times, they would be able to adjust their gestures with each repetition. The subjects were not allowed to see each other's performance, and only one subject and the authors were present in the studio during each recording session.

The sessions took place at the *Intermedia* video studio at the University of Oslo, featuring a blue screen background and high quality DV cameras. The cameras were placed in front and to the right of the subjects, at a distance of 4 meters. Firewire web-cams placed in the same positions allowed for rudimentary realtime video analysis, but it is the recordings from the DV-tapes that have been the source for our analysis.

Musical material. The musical material used for the studies were excerpts of piano music covering various playing techniques and styles:

1. Opening from Chopin's Scherzo no. 2 in Bb minor op. 31 (17 seconds) [22].
2. Opening from Scriabin's Sonata no. 5 op. 53 (10 seconds) [23].
3. Opening from the third movement of Beethoven's 3rd Piano Concerto (16 seconds) [24].
4. Opening from Messiaen's Regard des Anges from Vingt regards sur l'enfant Jesus (22 seconds) [25].
5. Excerpt from Tokyo 84 Encore by Keith Jarrett (16 seconds) [26].

The excerpts were chosen so as to present different features such as large pitch-space, salient phrases and attacks (excerpts 1 and 2), periodic and distinct textures (excerpt 3), percussive and dense textures (excerpt 4), and more groove-based types of textures (excerpt 5). The music was taken from commercially available CDs and DVDs, and recorded on one continuous track to facilitate playback and analysis. Each excerpt was repeated three times with 2 seconds of silence between similar excerpts and 5 seconds of silence before new excerpts.

Data display and analysis. The approximate nature of air playing (no keys to hit or miss, no fixed spatial coordinates), and the complexity of the gestures, makes it a formidable challenge to make reasonably well-founded judgments and analysis. Finding exact positions of hands and fingers in 3D from our video recordings seemed too difficult, and not particularly interesting, at this stage. Instead, we decided on an "eyes and ears" based annotation process.

As an aid in analysing the video material, we have developed the *Musical Gestures Toolbox*[1], a collection of patches built with the graphical music

---

[1] A beta-version is available at http://musicalgestures.uio.no

programming environment Max/MSP/Jitter [27]. Starting as a simple playback tool for video files, with image adjustments, rotation and zooming, it has grown to also include various types of motion-based analysis, sound analysis, preservation of musical pitch when changing playback speed, "posture"-recognition (figure 1), and automatic cropping. The latter function is particularly useful since it allows us to easily focus on various parts of the body, for example only the head or the hands. Also included are possibilities for saving snapshots and image sequences of the video stream (figure 2), and making comparative analysis of several video files (3).
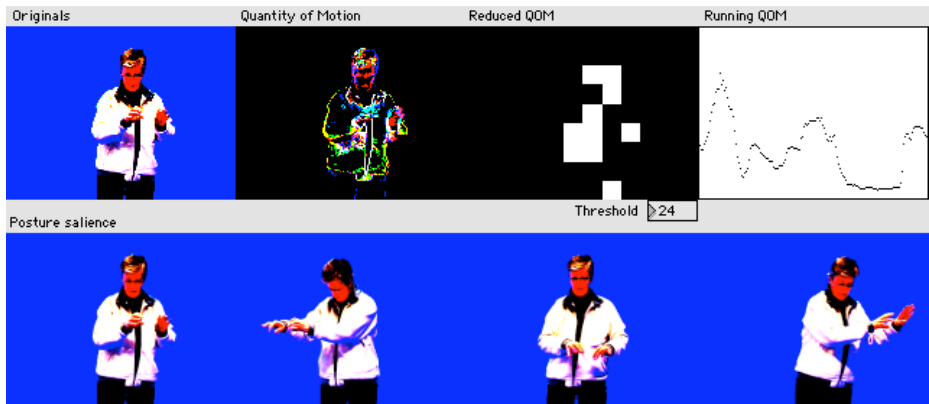
Fig. 1. Output of a patch made for storing an image every time the change in quantity of motion goes above a certain threshold. The original video stream and quantity of motion images in the top row, and the last four saved images below.

Also, using the *EyesWeb Motion Analysis Library*[2], we have looked at different types of movement features, such as the *silhouette motion image* (SMI) feature which creates trails of recent movements, and is an efficient tool for simulating the effect of short-term memory for trajectories, enhancing (or exaggerating) the contours of movements. A decay function allows for variable lengths of "lingering" and is useful for seeing gestures of pitch contours as well as accents (size of attack movements).

We have experimented with various other data collection techniques for gestures such as flex sensors, accelerometers, digitizing tablets, etc. but feel that the main challenge for the moment is to develop a better conceptual apparatus for dealing with sound-producing gestures and sound. Both gestures and sounds are continuous, yet making sense of gestures and sounds alike requires chunking continuous streams into units. Hence, conceptually we have a fundamental duality of the continuous and the discontinuous which we, for the moment, have simplified to a duality of *trajectories* and *postures*. Both elements can give us important

---

[2] See http://www.eyesweb.org for more information
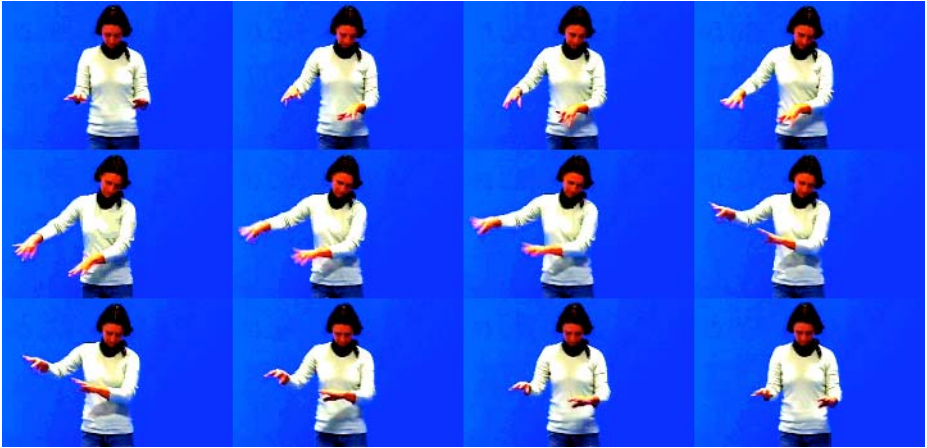
Fig. 2. Novice performer playing upward scales in the Scriabin excerpt. Although quite approximate, this example shows that there is a relatively good pitch-space to imagined keyboard correspondence (sequence running left to right, top row to bottom row).

Fig. 3. Output of a patch made for comparative analysis of three separate air piano performances, showing a novice, semi-expert and expert performer from left to right. The quantity of motion images with bounding boxes, are very useful when the movements are so subtle that they are difficult to see in the original video.

insights on gestures, as can be seen from figure 2 where the continuous trajectory is broken down into a series of snapshots. The postures can be understood as goal-points [21], i.e. as important points for evaluating the correspondence between sound-producing gestures and sound-events.

5   Gestural Correspondences

In evaluating air piano performances, we have taken as point of departure *the minimum necessary real sound-producing gestures by any pianist to generate the sound heard in the excerpts.* This means simply that in any real performance (i.e. not air performance) of the excerpts, keys have to be depressed by fingers in order to produce the sounds, and hands/arms have to move in order to position fingers so that they can depress the right keys. We use the term *correspondence* here to denote the relationship between what we can observe in the air playing and what would be the minimum movements necessary for any real performance of the excerpts. All correspondences we refer here are based on our subjects' fingers/hands/arms movements along an imagined keyboard (i.e. the horizontal axis) and onset motions by fingers/hands/arms (i.e. the vertical axis), and are ordered into the 7 categories of table 1.

In evaluating the air piano performances, we should note that although the subjects, prior to the video recording sessions, all stated that they understood the intentions of our air playing study, it is of course an open question to what extent they themselves would distinguish between sound-producing gestures and other more unspecific, yet music-related gestures such as head, torso, or whole body movements. It should also be noted that the lack of force feedback in air playing may have been awkward to some of the subjects, meaning that they would make different gestures playing air piano than they would playing the real thing.

Considering the intrinsically approximate nature of air playing, as well as the great difficulties we would have with a machine-based registration of sound-producing gestures mentioned earlier, we have chosen to give approximate, qualitative labels to the different degrees of correspondence between sounds and gestures that we have been able to observe. Using the various viewing tools mentioned in the previous section, we have carefully studied all the video recordings of the air playing gestures of our five subjects across the five different excerpts, but with the main focus on the last repetition of each excerpt (i.e. when the subjects had become most familiar with the music). By making detailed annotations, event-by-event, chunk-by-chunk, within each excerpt, we believe we have a fairly broad, distributed basis for our correspondence judgments. In making these judgments, we have also had a high degree of consensus amongst us (the authors).

We have chosen the following labels, and for convenience, assigned relative score values to the labels, to denote degrees of correspondence between air playing gestures and required real gestures:

- *No correspondence*, score value = 0, meaning the required sound-producing gestures are not visible.
- *Poor correspondence*, score value = 1, meaning the required sound-producing gestures are barely visible.
- *Approximate correspondence*, score value = 2, meaning the required sound-producing gestures are clearly visible, but inexact or wrong with regards to details.
- *Good correspondence*, score value = 3, meaning the required sound-producing gestures are clearly present and also match quite well in details.

Although these score values represent qualitative judgments of correspondences, we have for the sake of comparison calculated simple averages for each subject across the five excerpts used in our observation studies here, in order to make the summary of correspondences in table 1. It should be remembered that the five excerpts used were quite dissimilar, and they were deliberately chosen to expose the subjects to a variety of sound-producing gestures. Yet, there is still a fairly consistent level of performance by each of the subjects across the excerpts, seldom resulting in greater correspondence degree differences than 1.

Table 1. Correspondences of observable air playing gestures by all subjects (A–E), on a scale from 0–3, where 3 is good correspondence with the music. See text for details.

Feature	A	B	C	D	E
1. Overall activity correspondence, i.e. *density of gestures in relation to density of onsets in the music*, but regardless pitch and onset precision	1.4	1.8	2.6	2.6	3
2. Coarse pitch-space/keyboard-space correspondence, i.e. *relative locations of hands left-to-right on an imagined keyboard at phrase/section level*	0.8	1.4	2.0	2.4	2.8
3. Detail pitch-space/keyboard-space correspondence, i.e. *relative locations of fingers on an imagined keyboard at note-by-note level*	0.2	0.6	0.8	1.6	2.4
4. Coarse onset correspondence, i.e. *synchrony at downbeat or event level* (event in stead of downbeat in cases of less or non-periodic music)	1.6	1.4	1.8	2.6	2.6
5. Detail onset correspondence, i.e. *synchrony of finger and/or hand movements at note-to-note level*	1.0	0.2	0.8	1.8	2.2
6. Dynamics correspondence, i.e. *size and speed of hands/arms/body gestures in relation to loudness*	1.0	0.8	2.2	2.8	2.8
7. Articulation correspondence, i.e. *movements for accents, staccato, legato*, etc.	0.2	0.2	0.8	1.8	2.4

As for the categories we have designated here, the idea was to proceed from global to more detailed correspondences. Hence, in table 1, we start out with the overall activity correspondence, followed by pitch, onset, dynamics and articulation correspondences, hoping that this ordering should be informative as to how different levels of expertise are manifest in different aspects of air playing.

Category 1 concerns the overall activity correspondence, i.e. *the density of gestures in relation to the density of onsets in the music*, but regardless precision in onset-synchrony and pitch-space. This is a very coarse indication of the overall gestural activity in the air playing, and reflects the general or global impression of activity in the music such as *agitated, calm, fast, slow*, etc. Sometimes we could for example see a flurry of finger movements accompanying rapid, note-dense passages, which will give a rather good correspondence judgment for

overall activity, but poor values in terms of detail pitch and onset-synchrony. Interestingly, novices scored relatively well in this category.

The next two categories concern relative pitch-space correspondences. Category 2 indicates the *coarse* pitch-space to keyboard-space correspondence, i.e. *relative locations of hands left-to-right on an imagined keyboard at phrase/section level*. This implies a spatial resolution along the imagined keyboard at the octaves level, and reflects the relative register in relation to the entire piano keyboard at any given time. Some of the excerpts (Chopin, Scriabin, and Messiaen) were chosen for (amongst other features) this prominent use of large registers, and we can see that both novices and experts scored relatively well on this correspondence. However, with category 3, where the focus is on detail pitch-space to keyboard-space correspondence, i.e. *relative locations of fingers on an imagined keyboard at note-by-note level* (in most cases roughly within the octave ambit), we see that novices scored relatively lower than in categories 1 and 2, as did the experts, but relatively less so.

For onsets, we have made a similar distinction between coarse and detail correspondences. Category 4 indicates coarse onset correspondence, i.e. *synchrony at downbeat or event level* ("event" instead of downbeat in cases of less or non-periodic music, e.g. the Chopin, Scriabin, and Messiaen excerpts). The correspondence is relatively good for novices and experts alike, something we attribute to the salience of certain events in the Chopin, Scriabin, and Messiaen excerpts, and to the clear periodic nature of the Beethoven and Jarrett excerpts. As was the case for the pitch correspondences, the category 5 detail onset correspondence, i.e. *synchrony of finger and/or hand movements at note-to-note level*, shows on the whole less good correspondence than category 4 for both novices and experts.

Lastly, we were also interested in correspondences regarding dynamics and articulation. In category 6, we were looking for dynamics correspondence, i.e. *size and speed of hands/arms/body gestures in relation to loudness*, something that we believe is relatively well reflected in the gestures of both novices and experts. However, with category 7, articulation correspondence, i.e. *articulation movements for accents, staccato, legato*, etc., novices did not show much, but the experts tended to be quite clear about these kinds of movements.

Since the values in table 1 are based on qualitative judgments, and since we only had 5 subjects in this pilot study, we are reluctant to make more extensive correlation processing of these values. However, it seems reasonable to conclude that there is a continuum from novice to expert regarding overall, coarse correspondences between the music and sound-producing gestures: Novices clearly seem to perceive and make the corresponding gestures here. But for details in pitch, onsets, and articulations, i.e. what we would consider textural detail, novices seemed to make less and more inaccurate corresponding gestures.

6   Conclusions and Further Research

We understand air playing as motormimetic sketching, meaning that air playing includes the twin components of *imitative gestures* and *sketching*. Imitating the

gestures of others, in our case the innumerable gestures of musicians playing which we have seen throughout our lives, seems to be a resource for making sense of sounds. Although imitating sound-producing gestures may be a kind of "tacit" knowledge, we believe it is a resource that could be more actively exploited in both musicology and in various practical activities such as performance, composition, improvisation, and music education. However, this would require to acknowledge the value of sketching, i.e. of approximate, vague, "incorrect" gestures. This means to understand these gestural sketches as appropriate and useful global images of music, as playing an important role in parsing and chunking musical sound, as well as in grasping rhythmical, textural, melodic, and harmonic patterns. We thus believe it is a good idea to continue exploring air playing, as well as other sound tracing gestures. To do so, we also have to work towards the following:

- Enhanced means for gesture tracking, hopefully providing us with useful machine-generated data on movement trajectories.
- Enhanced conceptual and technical means for representing gesture trajectories and correlating these with sound.
- Better understanding of multimodal integration, in particular the neuro-cognitive bases for gesture-sound relationships.

References

1. A. S. Bregman. *Auditory Scene Analysis. The Perceptual Organization of Sound.* The MIT Press, Cambridge, Mass. & London, 1990.
2. R. I. Godøy and H. Jørgensen, editors. *Musical Imagery.* Swets and Zeitlinger, Lisse, 2001.
3. C. Cadoz. Musique, geste, technologie. In H. Genevois and R. de Vivo, editors, *Les nouveaux gestes de la musique*, pages 47–92. Editions Parenthèses, Marseille, 1999.
4. C. Cadoz and M. M. Wanderley. Gesture-music. In Marcelo Mortensen Wanderley and Marc Battier, editors, *Trends in Gestural Control of Music [CD-ROM].* IRCAM, Paris, 2001.
5. J. Haueisen and T. R. Knösche. Involuntary motor activity in pianists evoked by music perception. *Journal of Cognitive Neuroscience*, 13(6):786–792, 2001.
6. D. Sudnow. *Ways of the Hand.* Harvard University Press, Cambridge, Mass., 1978.
7. M. Mikumo. Encoding strategies for pitch information. In *Japanese Psychological Monographs No. 27.* The Japanese Psychological Association, 1998.
8. D. Rocchesso and F Fontana, editors. *The Sounding Object.* Edizioni di Mondo Estremo, Firenze, 2003.
9. A. M. Liberman and I. G. Mattingly. The motor theory of speech perception revised. *Cognition*, 21:1–36, 1985.
10. C. P. Browman and L. Goldstein. Articulatory gestures as phonological units. *Phonology*, 6:201–251, 1989.
11. L. Fadiga, L. Craighero, G. Buccino, and G. Rizzolatti. Speech listening specifically modulates the excitability of tongue muscles: a tms study. *European Journal of Neuroscience*, 15:399–402, 2002.

12. G. Hickok, B. Buchsbaum, C. Humphries, and T. Muftuler. Auditory-motor interaction revealed by fmri: Speech, music, and working memory. *Area Spt. Journal of Cognitive Neuroscience*, 15(5):673–682, 2003.
13. G. Rizzolatti and M. A. Arbib. Language within our grasp. *Trends in Neuroscience*, 21:188–194, 1998.
14. D. McNeill. *Hand and Mind: What Gestures Reveal About Thought*. University of Chicago Press, Chicago, IL, 1992.
15. S. Kita. How representational gestures help speaking. In David McNeill, editor, *Language and Gesture*, pages 162–185. Cambridge University Press, Cambridge, 2000.
16. A. Berthoz. *Le sens du mouevement*. Odile Jacob, Paris, 1997.
17. G. Rizzolatti, L. Fogassi, and Vittorio Gallese. Neurophysiological mechanisms underlying the understanding and imitation of action. *Nature Reviews Neuroscience*, 2:661—670, 2001.
18. M. Jeannerod. Neural simulation of action: A unifying mechanism for motor cognition. *Neuroimage*, 14:103–109, 2001.
19. J. Mandelbrojt. *Les cheveux de la réalité*. Editions Alliage (avec le soutien de la Fondation de France), Nice, 1991.
20. J. S. Gero and B. Tversky, editors. *Visual and Spatial Reasoning in Design. Key Centre of Design Computing and Cognition*. University of Sydney, 1999.
21. A. Wohlschläger, M. Gattis, and H. Bekkering. Action generation and action perception in imitation: an instance of the ideomotor principle. *Phil. Trans. R. Soc. Lond. B358*, pages 501–515, 2003.
22. F. Chopin. Scherzo no. 2 in B flat minor op. 31. Ivo Pogorelich, piano. Deutsche Grammophon 439 947-2.
23. A. Scriabin. Sonata no. 5 op. 53. Håkon Austbø, piano. Simax PSC 1055.
24. L. v. Beethoven. Beethoven Concertos pour piano 1 and 3. A la decouverte des Concertos. Francois-Rene Duchable, piano, John Nelson, conductor, Ensemble Orchestral de Paris [DVD]. Harmonia Mundi, 2003.
25. O. Messiaen. Regard des Anges, from Vingt regards sur l'enfant Jesus. Håkon Austbø, piano. Naxos 8.55089-30.
26. K. Jarrett. Tokyo '84 Encore. The Last Solo [DVD]. Image Entertainment, 1984.
27. A. R. Jensenius, R. I. Godøy, and M. M. Wanderley. Developing tools for studying musical gestures within the Max/MSP/Jitter environment. In *Proceedings of the International Music Computer Conference, Barcelona 5-9 September*, 2005.

Subject Interfaces: Measuring Bodily Activation During an Emotional Experience of Music

Antonio Camurri, Ginevra Castellano, Matteo Ricchetti, and Gualtiero Volpe

Infomus Lab, DIST, University of Genova,
Viale Causa 13, I-16145, Genova, Italy
`{toni, ginny, rmat, volpe}@infomus.dist.unige.it`
`http://infomus.dist.unige.it`

Abstract. This paper focuses on the relationship between emotions induced by musical stimuli and movement. A pilot experiment has been realized with the aim to verify whether there are correlations between the emotional characterization of music excerpts and human movement. Subjects were asked to move a laser pointer on a white wall in front of them while listening to musical excerpts classified with respect to the type of emotions they can induce.

Trajectories obtained moving the laser pointer have been recorded with a video camera and have been analyzed in a static and global way by using the EyesWeb platform. Results highlight a difference between trajectories associated to music stimuli classified as "fast" and "slow", in term of smoothness/angularity, suggesting the existence of a strong link between the emotional characterization of the musical excerpts listened to and the movement performed.

Subfield: expressive gesture and music.

Keywords: subject interfaces; emotion; expressive gesture; motor activation.

1 Introduction

Research in human-computer interaction more and more needs to take into account the communication aspect related to the "implicit channel", that is the channel through which the emotional domain interacts with the verbal aspect of communication (Cowie et al., 2001). In this context it is necessary to investigate how to communicate emotions to users and how to measure their emotional involvement.

Concerning the latter aim, understanding the nature of the emotional responses can help to appropriately develop emotion-oriented systems (Camurri et al., 2004a).

Several indicators can be taken into account to verify through which modalities an emotional phenomenon develops in human subjects: voice, facial expressions, physiological parameters, motor activation, etc. It seems necessary to consider emotional states as multimodal phenomena and to analyse also non-verbal aspects of emotional responses: psychological and neurophysiological research begin to show the importance of the movement component in characterizing an emotional process (Wallbott, 1998; Hillman et al., 2003; Berthoz and Viaud-Delmon, 1999).

The study presented in this paper focuses on motor activation as a component of an emotional process induced by musical stimuli. This investigation has been realized in collaboration with the group of Klaus Scherer (GERG, Geneva Emotion Research Group) of the Faculty of Psychology and Education Sciences of the University of Geneva in the framework of the EU-IST Network of Excellence HUMAINE (Human-Machine Interaction Network on Emotion).

The research investigates the *component-process model* of emotion of Klaus Scherer (Scherer, 1984, 2000; Scherer and Zentner 2001), that considers emotions as constantly changing phenomena integrating more components. Scherer suggests a model of emotion that takes into account a synchronization of the different emotion components. In this model, emotion is defined as a sequence of state changes in each of five organismic subsystems: the cognitive system (appraisal), the autonomic nervous system (arousal), the motor system (expression), the motivational system (action tendencies) and the experiential system (subjective feeling). The processes occurring in these five subsystems represent different components of an emotion: physiological arousal, motor activation, subjective feeling, action tendency (motivational component) and appraisal (cognitive component).

Our study focused on the motor activation component: a pilot experiment has been conducted aiming at verifying whether there are correlations between the emotional characterization of music excerpts and human movement, i.e. whether it is possible to refer to "expressive gesture" (Camurri et al., 2004a). The following sections will address the problem of measuring the emotional involvement in subjects, the performed experiment, the data analysis, and a discussion of the obtained results.

2   How to Measure an Emotional Experience: The Choice of the Laser Pointer

In our pilot experiment, music has been used as an emotion induction technique. To understand the relationship between music and emotion, it is necessary to investigate their time-varying relationships: continuous response methods allow one to record, during the listening process, the emotions induced by music without interruption.

In order to obtain continuous measures of emotion, several types of self-report techniques, devices and interfaces can be used. Scherer and colleagues (Scherer et al., 2002) suggested to use indicators (e.g. physiological recording, coding of non-verbal behaviour) other than verbal report, which may reflect inferences of emotional meaning rather than true reactions. Camurri and colleagues (Camurri et al., 2004a) performed an experiment in which participants indicated to what extent they were emotionally involved with the music by moving a MIDI-slider up and down. Schubert (Schubert, 2001, 2004a, 2004b) investigated a wide range of continuous measure devices used to record emotional response during listening to music. Our purpose was to find an adequate way to obtain measures of the emotional engagement, that were halfway between conscious (e.g. mouse, slider, haptic interfaces to communicate the experienced emotion) and unconscious (physiological measures: e.g., skin conductance, heart rate, breath rate, blood pressure (see Krumhansl, 1997a, 1997b); brain activity measures: e.g., EEG, PET, fMRI, magnetoencephalography (see Blood et al., 1999, 2001; Panksepp and Bernatzky, 2002)) conveyance. Our choice

considered subjects' motor activation as a channel to convey emotional involvement, and a laser pointer has been chosen as interface. During the experiment, subjects can move freely the laser pointer thus drawing with the laser dot on a white wall in front of them. The resulting trajectories of the laser dot are considered *expressive gestures* (Camurri et al., 2004a) communicating the experienced emotion.

The laser pointer is a simple interface to learn and to interact with, so it has been accepted very easily by subjects. Further, it has an interesting property: it plays the role of a sort of amplifier of small hand and arm movements. A few degrees of rotation or a small movement of the hand can be reflected in an ample spot variation on the wall. It is not requested to the subject to perform ample movements, which might result unnatural and difficult, but we rather aim at detecting small variations and perturbations of semi-conscious movements. We did not give any instruction to subjects: just to remain in the wall area in front of them. So, after some learning and tuning phase, subjects tended to move "semi-consciously" the laser pointer while concentrated only in listening. The amount of information contained in the laser trajectories may be surprisingly high: for example, geometric and repetitive patterns might imply a low emotional involvement of the subject (who is doing cognitive tasks, maybe distracted or bored by the music excerpt); sudden starts or stops, amount of changes in direction, etc. may reflect relevant moments of the emotional process going on during listening. Therefore, the performed pilot experiment suggests a new way to evaluate emotions induced by musical stimuli that overcomes the problem of verbal reports, by taking into account a non-verbal motor behaviour.

3   The Pilot Experiment

3.1   Subjects

A group of twenty people (nine male and eleven female) from twenty-one to thirty years old participated to the experiment. Seven subjects out of twenty are musician, sixteen listen to music more than two hours in a week and six have already been involved in experiments about music and movement.

3.2   Stimuli

A set of music excerpts provided by GERG and already evaluated by subjects in a previous experiment done at GERG. The music excerpts include repeated sections and have, on average, the same duration (about two minutes).

GERG selected eight musical excerpts classified with respect to the emotions that they evoke in subjects according to Scherer's eclectic approach (Scherer, 2003) and that, in turn, are grouped into four different classifications (fast positive, slow positive, fast negative, slow negative, with two excerpts for each group): Chopin, Concerto n°1, Romance, Larghetto, measures 13-37, duration 2:03 (Pleasant, Slow Positive); Mendelssohn, Trio pour piano n°1, Deuxième mouvement, measures 1-26, duration 1:59 (Pleasant, Slow Positive); Bruch, Kol Nidrei, Adagio pour violoncelle et orchestre avec harpe, measures 9-25, duration 1:54 (Sad, Slow Negative); Albinoni, Adagio en sol majeur, measures 1-30, duration 1:56 (Sad, Slow Negative); Milhaud, Scaramouche, III Brazileira, duration 2:08 (Funny, Fast Positive); Saint-Saëns, Le

Carnaval des animaux, Final, duration 1:52 (Funny, Fast Positive); Bartok, Sonate pour piano, BB88, Premier mouvement, measures 183-end, duration 1:23 (Aggressive, Fast Negative); Stravinsky, Le Sacre du Printemps, Sacrificial Dance, measures 135-end, duration 1:59 (Aggressive, Fast Negative).

Another music excerpt was used for a training session before starting the experiment: Rossini, La gazza ladra, Ouverture, measures 195-267, duration 1:06.

3.3   Set Up

The experiment was realized in a square room with soft lights, in order to allow one the detection of the laser's movement without compromising the visibility and in order to create a soft, comfortable and as most as possible natural environment. A computer with the audio files was connected to a Yamaha Digital Mixer 01v that in turn was connected to two Genelec loudspeakers. A video camera Panasonic GP KR222 with an s-video cable and 12.5 optics was used to record the movement of the laser. Constant shutter and 25fps non interlaced were used. The outputs of the video camera and of the audio mixer were connected to a DV recorder Sony GV-D300E in order synchronize audio and video. AVI files were obtained from the DV recordings.

3.4   Method

Subjects were asked to move a laser pointer on a white wall in front of them while listening to music excerpts characterized with respect to emotions they can induce. A training session preceded the experiment so that the subjects gained familiarity with the task and the experiment set up and environment. Each subject listened to four music excerpts: one slow positive, one slow negative, one fast positive, and one fast negative. The choice and the order of the excerpts were completely random. While listening to music, the trajectories performed by the subjects on the wall were recorded with a video camera (except during the training session).After listening to each excerpt, subjects compiled a questionnaire (provided by GERG). The questionnaire contains a group of labels chosen on the basis of the eclectic approach: subjects used the same labels that identify the excerpts.

From an analysis of the questionnaires it resulted that 65% of the subjects associated the characterizing label to all the excerpts they listened to, 15% only three times out of four, and 20% only two time out of four. At the end of the whole session each subject compiled a background questionnaire.

4   Analysis and Results

As a first hypothesis we assumed that the emotions induced by the musical stimuli in the subjects were really felt by them, accordingly to the questionnaire information.

The main objective was to verify if the music excerpts evoked measurable motor activity: it has been necessary to look for correlations among features of the trajectories performed by subjects with the laser pointer and emotional characterization of the music excerpts a subject has listened to. The study started from an analysis having a global but static nature. The laser trajectories were integrated over time, obtaining for each video file a bitmap summarizing the trajectory followed

during the whole listening. Therefore, each bitmap represents a graphical subject response (GSR) from the listening of a single music excerpt. The first step of the analysis consisted therefore of a **global analysis** considering such overall trajectories. This kind of analysis of the global trajectory patterns appears to be useful not only to verify which information one can obtain, but also because it provides an effective empiric approach to the analysis of the laser movement. In fact, this approach also provides the possibility of verifying which movement cues explain the behavior of subjects and, therefore, which cues it could be interesting to extract in a subsequent dynamic analysis. The following steps summarize the analysis.

4.1   Step 1: Extraction of Global Trajectories

By using the EyesWeb platform (Camurri et al., 2000a, 2000b, www.eyesweb.org) we obtained GSRs displaying the continuous path of the laser pointer (Fig. 1).
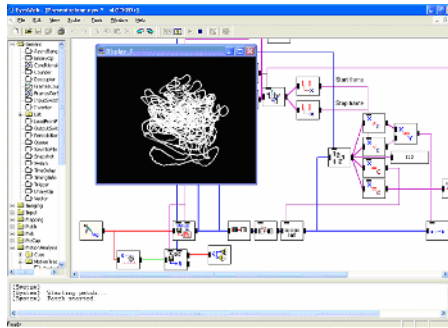

Fig. 1. The EyesWeb application for generating GSRs from subjects' laser movements

We obtained both a single, overall GSR for each listened musical excerpt, as well as multiple GSRs for each stimulus corresponding to the phrases composing the excerpt. In this preliminary analysis we started with the single overall GSR for each excerpt.

4.2   Step 2: Identification of Relevant Trajectory Features

The second step consisted of identifying a collection of descriptors to be employed for classification purposes. Such descriptors have to be related to specific features of the trajectory patterns. We identified the following features: angularity, rarefaction, spatial occupation, vertical symmetry, horizontal symmetry, central symmetry, compactness, lateral location, vertical location, angular tendency, and spatial extension. E.g., we can define angularity, rarefaction and compactness as follows:

- *Angularity*: Arccos of the angle between two successive segments of the trajectory
- *Rarefaction*: Density of the traced points: white pixels / total pixels in the bounding rectangle (i.e., the rectangle containing the whole trajectory pattern drawn by the laser pointer)
- *Compactness*: Use of the space by the subjects, e.g., if a subject use a portion of space (bounding rectangle) completely the pattern can be considered compact

4.3   Step 3: Providing Measures for Relevant Trajectory Features

Measures of the above mentioned trajectory features were obtained through a manual annotation. To this aim, it was necessary to define precise, clear, and unambiguous criteria for evaluating these features in a way as objective as possible. For manual annotation every pattern was evaluated with a value from 0 to 4 with respect to each specific feature: this range is usually adopted in the literature in similar cases.

4.4   Step 4: Evaluation of the Graphical Subjects' Responses

Five evaluators (different from the subjects participating to the listening experiment) performed manual annotation, in order to verify the degree of coherence among people. Average and variance of the five evaluations for each feature were calculated.

4.5   Step 5: Statistical Analysis

As a first step, a global overview of the data distribution was obtained. This preliminary qualitative analysis was useful for deciding how to perform a cluster analysis for grouping the patterns on the basis of similar values of the considered features. By observing the extracted GSRs, it appeared immediately evident that each subject had a well defined motor behaviour (Fig.2) and it is often possible to recognize a subject by looking at his/her GSRs.

Our aim was to find out whether an invariant behaviour going beyond the intrinsic subjectivity of each person could be extracted with a static and global analysis of the

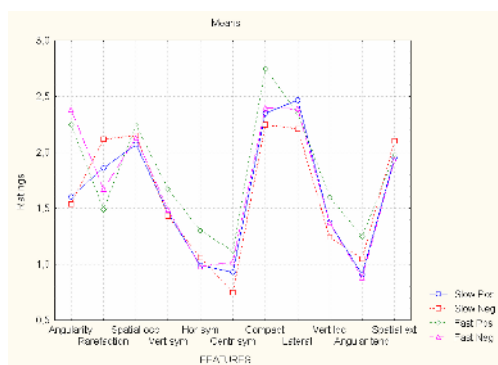**Fig. 2.** The four GSRs of a single subject

Fig. 3. The mean of all the ratings (performed by the five evaluators) of all the GSRs with respect to each feature for the four emotions

subjects' gestures. In order to verify this, the mean of all the ratings of all the features for the four emotions were calculated (see Fig. 3): each line in the figure is related to the emotional classification of the excerpts and each point on it represents the mean of all the ratings (performed by the five evaluators) of all the GSRs with respect to each feature.

From this qualitative analysis it was possible to observe that there are some features that seem to be richer of information with respect to others: they enable to distinguish among the patterns, and explain a likely general behaviour of the subjects. For example, the motor behaviour with respect to *angularity* is well defined in all the subjects: the global patterns related to the excerpts emotionally classified as "fast" show high values of angularity, whereas "slow" patterns show low values of angularity, so the correspondent trajectories are smooth. Concerning *rarefaction*, another result emerges: the slow patterns seem to be more rarefied than the fast ones; music excerpts classified as fast probably induce a movement having a higher velocity, so the motor activation is high. *Compactness* is another feature allowing one to distinguish among the patterns: it is evident a difference between "fast" and "slow" patterns, since the latter are less compact. One can conclude that the critical features are angularity, rarefaction, and compactness, where angularity seems to obtain the biggest separation between fast and slow patterns.

4.6 Step 6: Clustering Global Trajectories

A cluster analysis was carried out to verify whether it was possible grouping the subjects' patterns on the basis of the eleven evaluated features and whether there were correlations among the trajectory features and the emotional characterization of the music excerpts. This operation was performed on all the subjects' patterns. Cluster analysis was performed with an EyesWeb application running the K-Means algorithm. Three different cluster analyses were performed: each of them obtained, respectively, four, three and two clusters. All the cluster analyses were also carried out first with respect to all the eleven features and secondly according to only the critical features, that is the features identified above. The triple choice of seeing what happens with the creation of four, three and two clusters is due to the fact that it would be interesting to verify if the evaluated features allow one to distinguish among the four emotional characterization or between fast/slow and positive/ negative patterns only. After comparing the results of the different cluster analyses and verifying what is the best approach, we performed the second step, that is, to verify how the values of the features are distributed in the obtained clusters. After performing the three types of cluster analysis, subjects whose graphical responses were put in the same cluster were eliminated, because of the apparent lack of differentiation among the different graphical responses. These subjects in fact distinguish from the others because of their behaviour invariance. A comparison of the results achieved before the elimination of the invariant subjects and after it, showed that, with the elimination, one obtains a better subdivision, in terms of better percentages, of the GSRs graphical subjects' responses in the different clusters.

4.7 Results

After obtaining the clusters, the percentage of the emotional characterizations corresponding to them and the mean percentage value were calculated: the clustering

analysis highlighted that in all the cases of clustering the classification of the patterns improves considering only the critical features.

To determine the most promising cluster analysis, we considered the mean percentage values associated to each clustering operation. After a normalization taking into account the number of the clusters and comparing all the mean percentage values, it emerged that the cluster analysis grouping the patterns in two clusters on the basis of the three critical features gives the best results. In this case, between the two different classifications, fast/slow and positive/negative, the best classification is the latter (positive/negative): in this classification the cluster analysis obtains the higher degree of differentiation. Anyway, both the classifications were evaluated. The next step was to consider the values of the three features in correspondence of slow, fast, positive and negative patterns. Here we compared the behaviour of fast and slow patterns and the one of positive and negative patterns. Concerning the fast patterns, these ones are prevalent in cluster 2, whereas the slow patterns are more numerous in cluster 1.

Results show that the fast patterns in cluster 2 are very angular (80%), not rarefied (90%) and very compact (80%), whereas the slow patterns in cluster 1 are mainly not angular (50%, whereas the angular patterns are the 25% and the intermediate ones are the 25%), not rarefied (53.6%, rarefied 39.3% and intermediate 7.1%) and compact (64.3%, 28.6% not compact and 7.1% intermediate).

These results suggest that the fast patterns can be separated from the slow ones on the basis of the angularity only.

If one takes into account the positive/negative classification, the positive patterns prevail in cluster 1 and the negative ones in cluster 2. The negative patterns are angular (63.6%, not angular 27.3% and intermediate 9.1%), not rarefied (72.7%) and compact (63.6%). The positive patterns are angular (44.8%, not angular 38% and intermediate 17.2%), not rarefied (62.1 %, rarefied 24.1% and intermediate 13.8%) and compact (82.8%, not compact 10.3% and intermediate 6.9%).

These results suggest that, on the basis of these features, the positive and the negative patterns don't distinguish from each other.

Global results summarized in the following tables (Fig. 4 and Fig. 5).

CLUSTER 1	FEATURES		
EMOTION	Angularity	Rarefaction	Compactness
Slow	Low (50%)	Low (53.6%)	High (64.3%)
Positive	High (44.8%)	Low (62.1%)	High (82.8%)

Fig. 4. Percentages values of angularity, rarefaction and compactness in slow and positive patterns in cluster 1

CLUSTER 2	FEATURES		
EMOTION	Angularity	Rarefaction	Compactness
Fast	High (80%)	Low (90%)	High (80%)
Negative	High (63.6%)	Low (72.7%)	High (63.6%)

Fig. 5. Percentages values of angularity, rarefaction and compactness in fast and negative patterns in cluster 2

5 Discussion

The qualitative results achieved observing the graph representing the mean of all the ratings of all the GSRs' features (Fig. 3) led to hypotheses that have been confirmed only in part by the analysis. The graph of figure 3 shows a well distinct separation between fast and slow patterns on the basis of angularity, rarefaction and compactness. The richness of information contained in these features was checked and confirmed by the cluster analysis: this kind of analysis showed that the best degree of separation of the patterns is obtained when they are grouped according to these features. Nevertheless, the cluster analysis allows one to distinguish fast and slow patterns on the basis of the angularity only. Therefore, it is possible to argue that the fast patterns could be, on average, angular and the slow ones not angular.

One can conclude that subjects, moving the laser pointer, synchronize with the rhythm of the excerpts: if the velocity of the music increases, consequently the velocity of the arm movement increases as well as the direction changes frequently.

Concerning the differences between positive and negative patterns, the considered features are not able to differentiate between them.

Two conclusions can be drawn: either the choice of the features is, in this case, inadequate or the static analysis does not differentiate the different behaviour. First of all, some of the eleven features are certainly redundant since they do not allow one to obtain clusters that are coherent with respect to an emotional characterization. However, we kept the possibly redundant features in order to not lose information and evaluate at posteriori the relevance of each cue. Secondly, the loss of information due to the static analysis is another significant cause: the GSRs are considered in a temporal range comprising the whole subjects' performance.

6 Conclusions

6.1 Future Developments

This study aimed at investigating the link existing between music, emotions and movement. Music was used as an induction technique of emotions and motor activation of subjects was investigated with a pilot experiment using a laser pointer as means to communicate an emotional state in a continuous way. From a preliminary static analysis that focused its attention on the global patterns obtained by the subjects moving the laser pointer, it emerged that angularity is the feature that mainly explains the motor behavior: results show that graphical subjects' responses (GSRs) corresponding to fast music excerpts have a high angularity, whereas those ones associated to slow music excerpts are smooth. In practice, subjects, moving the laser pointer, seem to synchronize with the rhythm of the excerpts: the faster is the tempo of the piece of music, the faster is the movement performed, and this increases the frequency of direction changes. Therefore, it seems that there is a strong link between the emotional characterization of the excerpts listened to and the movement performed, in practice a sort of resonance between music and motor activation. These results are confirmed by previous studies (Popescu et al., 2004), in which it was demonstrated that, during music listening, activity in motor-related brain structures correlated with measures of rhythmicity derived from the music. Our analysis shows

that, statically, the features proposed to characterize the GSRs are redundant and only the evaluation of the angularity enable a distinction between the patterns. Another result is that static and global analysis does not allow one to distinguish between positive and negative patterns, but between fast and slow only. These results provide indications about the approaches to follow in the future. The static analysis of whole trajectories provides an effective empiric approach to the analysis of the laser movement. In fact, this approach provides the possibility of verifying which cues could be extracted in a dynamic analysis. The idea is to verify whether some features already taken into account during the static analysis but not critical with respect to distinction between the emotional characterizations of the patterns, are more significant during a dynamic analysis. In performing a dynamic and punctual analysis, the (x,y) coordinates of the laser can be obtained frame by frame; trajectories can be analyzed and features can be extracted with the support of EyesWeb Trajectory Analysis Library (Camurri et al., 2004b). It is possible to verify also how movement punctually changes with respect to time and how such changes can be correlated with the musical structure. The idea is then to work on the dynamic behaviour of a single subject, trying to find out which are the cues explaining the expressive gesture generated by each subject and verifying in a second step, at a higher level, if there is a common behaviour among the subjects, not only in terms of same characteristic cues, but also relatively to a possible same dynamic of the cues themselves.

6.2   The Need for a Novel Approach

The experiment we performed highlighted several critical problems. The choice of the laser pointer as subject interface and the high degrees of freedom left to subjects in order to have naturalness and emotionally rich movements implies a high degree of variance in the motor behaviour of the subjects. The comparison at a low level demonstrated a high degree of subjectivity with consequent low evidence of a common behaviour among subjects. A new methodological approach aiming at evaluating the motor behaviour of the subjects seems needed: the idea is to work on the analysis of the response of the single subject and try to find analogies within the same subject. In a second phase, it should be searched analogies among subjects in a-posteriori and at a higher level of abstraction, in terms of dynamic profile of higher-level motor features. At different stages of the research it is necessary to understand the influence of the subjects' profile on the obtained results in terms of correlation between stimuli and responses. Psychological profiles could be strategic in order to find inter-subject analogies and, at the same time, could contribute to a general theory. Moreover, in the definition of the relationship between music and emotion, there is the need to develop dynamic models explaining the dynamics of an emotional response. It seems necessary to try to find out whether there are correlations between the temporal profile of the stimulus and the vitality affect (changes in the temporal dynamic of an emotion).

Acknowledgements

Scherer and Marcel Zentner (GERG, University of Geneve) for their contributes to the experiment mentioned in the paper, described in detail in a paper in preparation.

References

1. Berthoz, A., Viaud-Delmon, I., (1999), "Multisensory integration in spatial orientation", Current Opinion in Neurobiology 1999, 9:708-712.
2. Blood, A.J., Zatorre, R.J., Bermudez, P., and Evans, A.C., (1999), "Emotional responses to pleasant and unpleasant music correlate with activity in paralimbic brain regions", Nature Neuroscience, vol. 2 no. 4.
3. Blood, A.J., and Zatorre R.J., (2001), "Intensely pleasurable responses to music correlate with activity in brain regions implicated in reward and emotion", PNAS, vol.98 no. 20.
4. Camurri, A., Hashimoto, S., Ricchetti, M., Trocca, R., Suzuki, K., and Volpe G., (2000a) "EyesWeb -Toward Gesture and Affect Recognition in Interactive Dance and Music Systems", Computer Music Journal, 24:1, pp. 57-69, MIT Press, Spring 2000.
5. Camurri, A., Coletta, P., Peri, M., Ricchetti, M., Ricci, A., Trocca, R., and Volpe G., (2000b), "A real time platform for interactive dance and music systems", 2000.
6. Camurri, A., Mazzarino, B., Ricchetti, M., Timmers, R., and G. Volpe, (2004a),"Multimodal analysis of expressive gesture in music and dance performances", in A. Camurri, G. Volpe (Eds.), "Gesture-based Communication in Human-Computer Interaction", LNAI 2915, Springer Verlag, 2004.
7. Camurri, A., Mazzarino, B., and Volpe, G., (2004b), "Analysis of Expressive Gesture: The Eyesweb Expressive Gesture Processing Library", in A. Camurri, G.Volpe (Eds.), "Gesture-based Communication in Human-Computer Interaction ", LNAI 2915, Springer Verlag, 2004.
8. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W. and Taylor, J.G., (2001), "Emotion recognition in human-computer interaction", IEEE Signal Processing Magazine, January 2001.
9. Hillman, C.H., Rosengren, K.S., and Smith, D.P., (2003), "Emotion and motivated behavior: postural adjustments to affective picture viewing", Biological Psychology 66 (2004) 51-62.
10. Krumhansl, C.L. (1997a), "An exploratory study of musical emotions and psychophysiology", Canadian Journal of Experimental Psychology, 51:4, 336-352.
11. Krumhansl, C.L. (1997b), "Psychophysiology of musical emotions", Proceedings ICMC.
12. Panksepp, J., Bernatzky, G. (2002),"Emotional sounds and the brain: the neuro-affective foundations of musical appreciation", Behavioral processes 60, 133-155.
13. Popescu, M., Otsuka, A., and Ioannides A.A., (2004), "Dynamics of brain activity in motor and frontal cortical areas during music listening: a magnetoencephalographic study", NeuroImage 21 (2004) 1622-1638.
14. Scherer, K.R., (1984), "On the nature and function of emotion: a component process approach", in K.R. Scherer & P. Ekman (Eds.), Approaches to emotion (pp.293-317). Hillsdale, NJ: Erlbaum.
15. Scherer, K.R., (2000), "Emotions as episodes of subsystem synchronization driven by nonlinear appraisal processes", in Lewis, M. & Granic, I. (Eds.) Emotion, Development, and Self-Organization (pp. 70-99). New York/Cambridge: Cambridge University Press.
16. Scherer K.R., Zentner M.R., (2001), "Emotional effects of music: production rules", In P.N. Juslin & J.A.Sloboda (Eds). Music and emotion: Theory and research (pp. 361-392). Oxford: Oxford University Press.

17. Scherer K.R., Zentner M.R. and Schacht A., (2002),"Emotional states generated by music: an exploratory study of music experts", Musicae Scientiae, Special Issue 2001-2002, 149-171.

18. Scherer K.R., (2003), "Why music does not produce basic emotions: pleading for a new approach to measuring the emotional effects of music", Proceedings of the Stockholm Music Acoustics Conference, August 6-9, 2003 (SMAC 03), Stockholm, Sweden.

19. Schubert E., "Continuous measurement of self-report emotional response to music", (2001), in P. Juslin and J. Sloboda (Eds.), Music and emotion: theory and research, Oxford 2001 University Press, Oxford, pp.393-414.

20. Schubert, E., "Modeling perceived emotion with continuous musical features", (2004a), Music Perception, Summer 2004, vol.21, no.4, 561-585.

21. Schubert, E., "EmotionFace: prototype facial expression display of emotion in music", (2004b) Proceedings of ICAD 04-Tenth Meeting of the International Conference on Auditory Display, Sydney, Australia, July 6-9, 2004.

22. Wallbott, H.G., (1998), "Bodily expression of emotion", European Journal of Social Psychology, Eur. J. Soc. Psychol. 28, 879-896.

From Acoustic Cues to an Expressive Agent

Maurizio Mancini[1], Roberto Bresin[2], and Catherine Pelachaud[1]

[1] LINC, IUT de Montreuil, University of Paris8
m.mancini@iut.univ-paris8.fr, c.pelachaud@iut.univ-paris8.fr
http://www.iut.univ-paris8.fr/greta
[2] Royal Institute of Technology, Stockholm
roberto@speech.kth.se
http://www.speech.kth.se/~roberto

Abstract. This work proposes a new way for providing feedback to expressivity in music performance. Starting from studies on the expressivity of music performance we developed a system in which a visual feedback is given to the user using a graphical representation of a human face. The first part of the system, previously developed by researchers at KTH Stockholm and at the University of Uppsala, allows the real-time extraction and analysis of acoustic cues from the music performance. Cues extracted are: sound level, tempo, articulation, attack time, and spectrum energy. From these cues the system provides an high level interpretation of the emotional intention of the performer which will be classified into one basic emotion, such as happiness, sadness, or anger. We have implemented an interface between that system and the embodied conversational agent Greta, developed at the University of Rome "La Sapienza" and "University of Paris 8". We model expressivity of the facial animation of the agent with a set of six dimensions that characterize the manner of behavior execution. In this paper we will first describe a mapping between the acoustic cues and the expressivity dimensions of the face. Then we will show how to determine the facial expression corresponding to the emotional intention resulting from the acoustic analysis, using music sound level and tempo characteristics to control the intensity and the temporal variation of muscular activation.

1 Introduction

Listening to music is an everyday experience. But why do we do it? For example one could do it for tuning one own mood. In fact research results show that not only we are able to recognize different emotional intentions used by musicians or speakers [1] but also we feel these emotions. It has been demonstrated by psychophysical experiments that people listening to music evoking emotions experience a change in biophysical cues (such as blood pressure, etc.) that correspond to the feeling of that specific emotion and not only to the recognition [2].

What happens when it is a computer listening to the music? In HCI applications affective communication plays an increasingly important role [3,4]. It would be helpful if agents could express what they perceive and communicate it to the human operator.

Animated agents are a new paradigm for human-machine interfaces. They are entities with a human-like appearance capable of taking part in dialogs with users. They communicate complex information through several modalities: voice, intonation, gaze, gesture, facial expressions, etc. They are used in a variety of applications for their ability to convey complex information in a quite human-like manner, that is through verbal and nonverbal communication [5, 6, 7, 8, 9, 10, 11].

We present a real-time application of an expressive virtual agent's face displaying emotional behaviors in relation with expressive music. As the quality of the music changes, the associated facial expressions varies. Emotions are mainly displayed on the face through facial expressions, but another important factor of the communication is how an expression is executed. Indeed a facial expression is defined by the signals (e.g., raised eyebrows and head nod) that composed it as well as their evolution through time.

In a previous research [12], we have developed a computational model of gesture expressivity. Six dimensions have been defined and implemented. In this paper we extend this work to the face. That is we provide a model to modulate facial animation based on expressivity qualifiers.

In the next section we will present other systems in which visual feedback of music performance is given through the animation of an virtual agent. Then we will describe the CUEX system and the set of acoustic tone parameters that are extracted from an expressive music performance. In the following, these tone parameters are called acoustic cues. We will then turn our attention toward the Greta virtual agent system. Section 4 describes the animation system while Section 4.1 presents the six dimension of expressivity we consider. Next, in Section we will expose our real-time application and we will provide information on the mapping between acoustic and animation parameters. Finally we will conclude the paper.

2   State of the Art

During the last 15 years, Japanese researchers have devoted great effort in the so called KANSEI information processing [13]. This has been suggested as a new kind of information technology to be investigated in parallel to the information processing of physical signals (e.g. audio signals) and of symbolic descriptions (e.g. text or music scores). Today, KANSEI information processing is applied to emotional and human communication including visual art, design and music, and it is also know as Affective Computing [14]. In more recent years research in the field of multimodal communication has shown a growing interest. In particular, in the field of music performance, it has been developed applications in which avatars can play instruments [15], dancers can establish a multimodal communication with robots [16] and embodied conversational agents interact with the user for example in a virtual theater [11]. More recently, Taylor and co-workers [17] designed an application in which acoustic cues (such as pitch, sound level, vocal timbre, and chord information) are used for driving the behavior of a synthetic virtual character. For example, this synthetic character could

turn his head towards the sound source more or less rapidly depending on the characteristic of the timbre. This application is similar to that proposed in this paper, in which more than one acoustic cue is mapped into the behavior of an expressive animated human-like character.

3   From Acoustic Cues to Emotions

CUEX (CUe EXtraction) is a system developed at KTH and Uppsala University for extracting acoustic cues from an expressive music performance [18] [19]. These cues can be mapped into a 2-dimensional space that represents the expressivity of the performance. For example the 2-dimensional space can be the pleasure-displeasure and degree of arousal space proposed by Russell [20].

Acoustic cues that can be extracted by CUEX are articulation (*legato* or *staccato*), local tempo (number of events in a given time window), sound level, spectrum energy, and attack time. Research in music performance has shown that musicians use acoustic cues in a particular way and combination in order to communicate emotions when playing [21] [22]. Particular combinations and relative values of the cues correspond to specific emotions. In 1 we present the use of acoustic cues by musicians when performing for communicating happiness, anger, or sadness as reported by Juslin [22].

In the system used in the present work, acoustic cues are mapped to an "emotional space" using a fuzzy logic approach [23]. For example, if a piece of music is played with legato articulation, soft sound level, and slow tempo it will

Table 1. Musicians' use of acoustic cues when communicating emotion in music performance (from [22])

Emotion	Acoustic cues	Emotion	Acoustic cues
Sadness	slow mean tempo	Anger	fast mean tempo
	large timing variations		small tempo variability
	low sound level		high sound level
	legato articulation		staccato articulation
	small articulation variability		spectral noise
	soft duration contrasts		sharp duration contrasts
	dull timbre		sharp timbre
	slow tone attacks		abrupt tone attacks
	flat micro-intonation		accent on unstable notes
	slow vibrato		large vibrato extent
	final ritardando		no ritardando
Happiness	fast mean tempo		rising micro-intonation
	small tempo variability		fast tone attacks
	small timing variations		bright timbre
	high sound level		sharp duration contrasts
	little sound level variability		staccato articulation
	large articulation variability		

be classified as "sad", while it will be classified as "happy" if the performance is characterized by a more staccato articulation, louder sound level, and faster tempo.

For the purpose of the present work, a real time version of CUEX based on pd is used. CUEX analyzes an input audio signal (a sound file or an audio stream) and provides the acoustic cues in output. A second pd patch takes these cues in input and maps them into emotions. In a recent application called the "ExpressiBall" both cues and emotions are visualized using a 3-dimensional object moving on the computer screen [24]. The object, a ball, changes position, size, and shape according to the acoustic cues, while the colour corresponds to the current emotion 1 [25]. The main idea of ExpressiBall is to provide music students with non-verbal and informative feedback while they practice to play with expressivity.

A recent review of 104 studies of vocal expression and 41 studies of music performance [1] reveals strong similarities between the two channels (voice and music performance) in the use of acoustic cues for communicating emotions. This could explain why listeners perceive music expressing emotions. This result opens for the experiment in the present work, i.e. to provide visual feedback to music performance by means of an expressive agent.
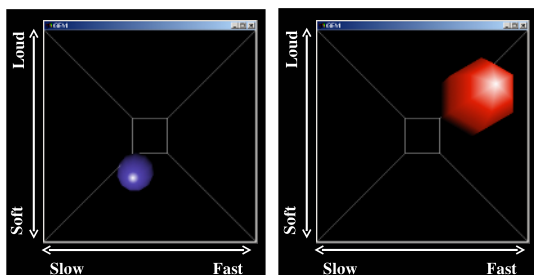

Fig. 1. The ExpressiBall: graphical interface for real-time feedback of expressive music performance. Dimensions used in the interface are: X = tempo, Y =sound pressure level, Z =spectrum (attack time & spectrum energy), Shape = articulation, Colour = emotion. Left figure shows the feedback for a sad performance. Right figure shows the feedback for an angry performance.

4   Real-Time FeedBack Agent

In this section we describe the virtual agent system we use as visual feedback for the acoustic cues extraction system CUEX. Our agent system, Greta, is an embodied representation of a virtual human able to communicate verbal and non-verbal behaviors (such as facial expressions, gaze, head movement) [26]. Based on theoretical studies [27, 28], we have developed a set of *modifiers* that modulates movement quality of a specific behavior [12]. We refer to the manner of behavioral execution with the term *expressivity* (see section 4.1 for a description of our set of modifiers).

The agent system linked to CUEX ought to provide continuous feedback to the user; it needs to produce animation in real-time where the animation parameters vary depending on the acoustic cues (see section 3). A schematic representation of our system is given in figure 2.
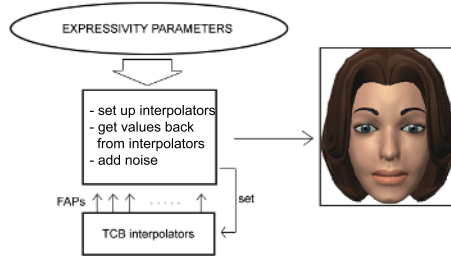

Fig. 2. The Greta's architecture

For the current application, the animation of the agent is controlled by specifying which signals to perform. These signals are dynamically altered by a set of *expressivity parameters* (see 4.1). Examples of signals are:

- *move head*: the agent changes head direction. A new target orientation for the head (corresponding to a given rotation around the x and/or y axis) is specified. The head reaches it and finally turns back to its original position;
- *blink*: the agent performs an eye blink;
- *assume expression (e)*: the agent shows the specified facial expression e.

To avoid a frozen agent during idle time, a function generates Perlin Noise [29] for the head orientation. This function is also applied to the head orientation during activity intervals creating slightly changed positions.

The Greta system calculates on-the-fly the appearance of the agent's face at given time. Concurrently it decides if any of the possible signals has to be performed or not. It also instantiates eye blinks. The action of performing eye blink is needed to simulate the biological need of keeping the eye wet. This kind of blinks appears every 4.8 seconds in average [30]. But this temporal characteristics varies depending on the actual emotion. For example the frequency of execution of blinks will decrease for emotions like sadness while it will increase during anxiety [31].

4.1   Greta's Expressivity

The Greta's behavior expressivity is parameterized by a set of *dimensions of expressivity* [12]. They modify the animation of the agent qualitatively. The 6 dimensions are:

- *Overall activation*: amount of activity (e.g., passive / static or animated / engaged), number of movements per time (head movement, gesture).

- *Spatial extent*: amplitude of movements (e.g., amplitude of raised eyebrow and head rotation). This parameter is somehow linked to the quantity of muscular contraction. It also increases/decreases the amplitude of head rotations.
- *Temporal extent*: duration of movements (e.g., quick versus sustained actions). Low values produce movements that last very shortly (e.g., the head quickly returns to the rest position) while higher values produce longer ones (e.g., the head remains rotated for some longer time, then it returns).
- *Fluidity*: smoothness and continuity of movement (e.g., smooth, graceful versus sudden, jerky). This parameter acts on the continuity between consecutive facial expressions and head movements. Figure 3 shows two examples of different settings for the fluidity parameter.
- *Power*: dynamic properties of the movement (e.g., weak / relaxed versus strong / tense). Higher / lower values increases / decreases the acceleration of movements, making movements become more or less powerful. Figure 4 shows some examples of curves with different tensions.
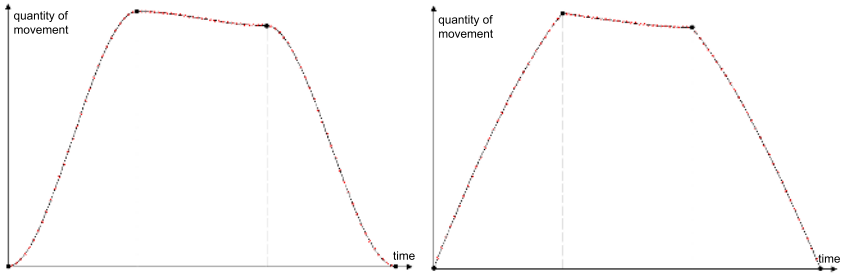- *Repetitivity*: this factor allows one to repeat the same expression several time in a row.

Fig. 3. Fluidity variation: left diagram represents normal fluidity, right diagram represents low fluidity for the same behavior
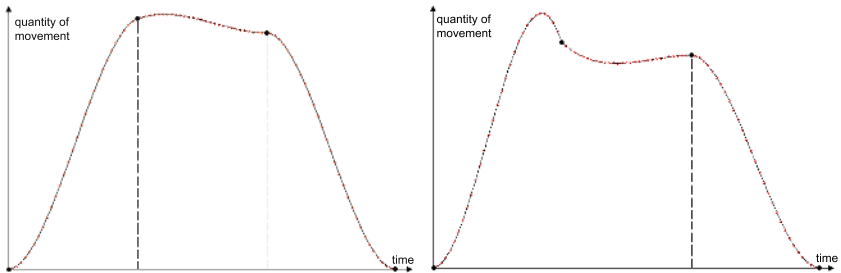

Fig. 4. Tension variation: left diagram represents low tension, right diagram represents high tension for the same keypoint animation with the presence of overshooting before the reaching of the fist keyframe

5   The Music2Greta FeedBack System

The music feedback system uses an agent as visual representation of the communicated music expression. It has been realized by interfacing the output of the acoustic features extraction system CUEX described in section 3 with the input of the Greta system described in section 4. In the next section we describe the structure of the system and the mapping between acoustic features and expressivity parameters. Figure 6 is an example of a possible sequence of animation frames produced by the system.

As shown in figure 5 our Music2Greta system is composed by the CUEX system and the Greta agent that communicate through a TCP/IP socket. Acoustic cues resulting from the CUEX elaboration are transmitted to Greta in real-time. The *Mapping* module receives them and calculates (see next section) the values to assign to the Greta's expressivity parameters.

What we are focusing on here is the *Mapping* module in figure 5. Table 2 sumarizes the correspondences we implemented inside this module.
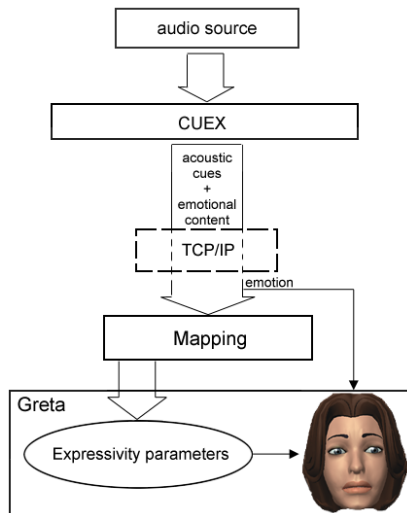

Fig. 5. Music2Greta architecture

Table 2. Parameters correspondence between CUEX and Greta

CUEX	Greta
1. Emotional intention	Face expression
2. Emotional intention	All parameters
3. Volume	Spatial & Power
4. Tempo	Temporal & Overall activation
5. Articulation	Fluidity

Fig. 6. This sequence shows an example of the output of the Music2Greta feedback system. From left to right and top to bottom we can see the agent rotating her head to look up-left. Then the expression changes from joy to sadness while the head rotates down-right.

As shown in lines **1** and **2** the emotional intention affects both the emotion shown by the agent and its expressive parameters (that is the way the agent will modulate its behavior). Then there is a correspondence between the acoustic cues extracted by CUEX and the agent's expressivity parameters. The next sections focus on each of these correspondences.

5.1   Emotion Intention to Facial Expressions

At every system clock, the CUEX system determines to which emotion the music corresponds to: happiness, anger, or sadness. The emotion with the highest value is selected. This information directly influences the emotion shown by the agent by the execution of an *assumed expression* signal. A representation of the emotion is retrieved from the face expression library of the agent and applied to the graphical representation of the face. The expression is hold until the perceived emotive content changes and another intended emotion is sent to Greta system. Information on emotional content also affects the way in which the agent blinks (for example, high emotive emotions like anger will increase the frequency of blink; sadness will decrease them).

5.2   Emotion Intention to Expressivity Parameters

Many research work focused on the influences that emotions have on the physical behavior of humans, not only on the the facial expression. Boone & Cunningham

[32] observed how children perceive emotion in expressive body movements and highlighted that for example slow movements are usually recognized by adults as a sadness emotional state. Abrillian et al. [33] analyzed multimodal behavior from a video corpus made of TV clips with emotive content. Wallbott studies [27] on non-verbal communication of acted emotions revealed that for example hot anger emotion causes an increasing of movement activity/expansivity and movement power/energy; elated joy .... and sadness ...

In the Greta2Music system, we apply similar mapping between emotions and expressivity parameters as reported in Wallbott [27]:

- Anger: agent's face becomes reddish and head moves towards the user. Eye blink frequency increases [31]. *Temporal, spatial, power* parameters are increased and *fluidity* is decreased. The result of these modifications is that the head's animation looks faster, more wide and less smooth.
- *Sadness*: agent's face becomes pale, head moves backward and looks mainly downwards. Eye blink frequency decreases [31]. *Temporal, spatial, power* parameters are decreased and *fluidity* is increased. That is the head's animation appears to be slower, less wide and more smooth.
- *Joy*: head looks towards the user and *spatial, power* and *fluidity* parameters are increased. So the head's animation results in a more wide, energetic and fluid movement.

5.3   From Acoustic Cues to Expressivity Parameters

The mapping from acoustic cues to expressivity parameters is inspired by results of a recent study on musician gestures in emotional expressive performance [34][35]. Dahl and Friberg found that musicians' body movements are correlated with their emotional intention. In particular viewers could recognize the emotional intentions just by watching video recordings of performing musicians without listening to the audio channel. The observed main characteristics of musicians' body movements for four emotional intentions were:

- *Happy*: normal movements, normal fluency, normal speed, and normal overall amount of movement
- *Angry*: irregular movements, jerky fluency, fast speed, and larger overall amount of movement
- *Sad*: more regular movements, smooth fluency, slow speed, and smaller overall amount of movement
- *Fearful*: irregular movements, jerky fluency, fast speed, and smaller overall amount of movement

Keeping in mind these observations as starting point, the extracted acoustic cues, namely sound level, tempo and articulation, are linearly mapped into the corresponding expressivity parameters using simple scaling to adapt their ranges of variation:

- *Volume*: the current sound level of the music performance is linearly mapped into the *Spatial* and *Power* expressivity parameters. E.g., it will influence

the degree of rotation of head movements as well as their acceleration and quantity of overshooting.

- *Tempo*: it represents the speed of the musical performance and influences the *Temporal* and *Overall activation* expressivity parameters. The first effect will be to vary the duration of head movements, the second will be to increase/decrease the frequency of head movements.
- *Articulation*: it reflects the style and the quantity of the articulation in the music performance, i.e. the amount of *staccato* or *legato*. It will vary the fluidity dimension. For example it will act on the continuity of the head movements making them less continuous and less co-articulated as the articulation becomes more *staccato*.

6   Conclusions

In this paper we have presented a system that provides real-time expressive visual feedback of music performance. As the music expressivity changes the facial expressions and the quality of movement of the agent get modified accordingly.

We aim at evaluating the system to test if the perception of expressive variations in music performance is enhanced using the agent as visual feedback. We are also interested in looking at what happens when there is a mismatch between musical quality and facial expression and behavior of the agent.

Further improvements to the system are also foreseen. The computation of the intermediate expression between 2 emotions will provide more consistent results by implementing the emotion representation along the activation-evaluation space, based on the work of [36]. The interpolation between 2 emotions will correspond to a curve in space where each point in space may correspond to an emotion for which a facial expression can be computed [37].

Acknowledgments

References

1. Juslin, P.N., Laukka, P.: Communication of emotions in vocal expression and music performance: Different channels, same code? Psychological Bulletin **129** (2003) 770–814
2. Krumhansl, C.L.: An exploratory study of musical emotions and psychophysiology. Canadian Journal pf Experimental Psychology **51** (1997) 336–352
3. Höök, K., Bullock, A., Paiva, A., vala, M., Chaves, R., Prada, R.: FantasyA and SenToy. In: Proceedings of the conference on Human factors in computing systems, Ft. Lauderdale, Florida, USA, ACM Press (2003) 804–805

4. Marsella, S., Johnson, W., LaBore, C.: Interactive pedagogical drama for health interventions. In: 11th International Conference on Artificial Intelligence in Education, Sydney, Australia (2003)

5. Gratch, J., Marsella, S.: Some lessons for emotion psychology for the design of lifelike characters. Journal of Applied Artificial Intelligence (special issue on Educational Agents - Beyond Virtual Tutors) **19** (2005) 215–233

6. Johnson, W., Vilhjalmsson, H., Marsella, S.: Serious games for language learning: How much game, how much AI? In: 12th International Conference on Artificial Intelligence in Education, Amsterdam, The Netherlands (2005)

7. Craig, S., Graesser, A., Sullins, J., Gholson, B.: Affect and learning: An exploratory look into the role of affect in learning with AutoTutor. Journal of Educational Media **29** (2004) 241–250

8. Massaro, D., Beskow, J., Cohen, M., Fry, C., Rodriquez, T.: Picture my voice: Audio to visual speech synthesis using artificial neural networks. In: International Conference on Auditory-Visual Speech Processing AVSP'99, Santa Cruz, USA (1999)

9. Bickmore, T., Cassell, J.: Social dialogue with embodied conversational agents. In J. van Kuppevelt, L. Dybkjaer, N.B., ed.: Advances in Natural, Multimodal Dialogue Systems. Kluwer Academic, New York (2005)

10. Kopp, S., Wachsmuth, I.: Synthesizing multimodal utterances for conversational agents. The Journal of Computer Animation and Virtual Worlds **15** (2004)

11. Nijholt, A., Heylen, D.: Multimodal communication in inhabited virtual environments. International Journal of Speech Technology **5** (2002) 343–354

12. Hartmann, B., Mancini, M., Pelachaud, C.: Implementing expressive gesture synthesis for embodied conversational agents. In: The 6th International Workshop on Gesture in Human-Computer Interaction and Simulation, Valoria, UniversitT de Bretagne Sud, France (2005)

13. Hashimoto, S.: Kansei as the third target of information processing and related topics in japan. In: Proceedings of KANSEI - The Technology of Emotion AIMI International Workshop, Genova (1997) 101–104

14. Picard, R.W.: Affective Computing. MIT Press, Cambridge (1997)

15. Lokki, T., S.L.V.R.H.J., Takala, T.: Creating interactive virtual auditory environments. IEEE Computer Graphics and Applications, special issue "Virtual Worlds, Real Sounds" **22** (2002)

16. A. Camurri, P. Coletta, M.R.G.V.: Expressiveness and physicality in interaction. Journal of New Music Research **29** (2000) 187–198

17. Taylor R., Torres D., B.P.: Using music to interact with a virtual character. In: 5th International Conference on New Interface for Musical Expression - NIME, Vancouver, Canada (2005) 220–223

18. Friberg, A., Schoonderwaldt, E., Juslin, P.N., Bresin, R.: Automatic real-time extraction of musical expression. In: International Computer Music Conference - ICMC 2002, San Francisco, International Computer Music Association (2002) 365–367

19. Friberg, A., Schoonderwaldt, E., Juslin, P.N.: Cuex: An algorithm for extracting expressive tone variables from audio recordings. Accepted for publication in Acoustica united with Acta Acoustica (2005)

20. Russell, J.A.: A circumplex model of affect. Journal of Personality and Social Psycholog **39** (1980) 1161–1178

21. Gabrielsson, A., Juslin, P.N.: Emotional expression in music. In Goldsmith, H.H., Davidson, R.J., Scherer, K.R., eds.: Music and emotion: Theory and research. Oxford University Press, New York (2003) 503–534

22. Juslin, P.N.: Communicating emotion in music performance: A review and a theoretical framework. In Juslin, P.N., Sloboda, J.A., eds.: Music and emotion: Theory and research. Oxford University Press, New York (2001) 305–333

23. Friberg, A.: A fuzzy analyzer of emotional expression in music performance and body motion. In: Proceedings of Music and Music Science, Stockholm 2004 (2005)

24. Bresin, R., Juslin, P.N.: Rating expressive music performance with colours. Manuscript submitted for publication (2005)

25. Bresin, R.: What is the color of that music performance? In: International Computer Music Conference - ICMC2005, Barcelona, ICMA (2005) 367–370

26. Pelachaud, C., Bilvi, M.: Computational model of believable conversational agents. In Huget, M.P., ed.: Communication in Multiagent Systems. Volume 2650 of Lecture Notes in Computer Science. Springer-Verlag (2003) 300–317

27. Wallbott, H.G.: Bodily expression of emotion. European Journal of Social Psychology **28** (1998) 879–896

28. Gallaher, P.: Individual differences in nonverbal behavior: Dimensions of style. Journal of Personality and Social Psychology **63** (1992) 133145

29. Perlin, K.: Noise, hypertexture, antialiasing and gesture. In Ebert, D., ed.: D. Ebert, editor, Texture and Modeling, A Procedural Approach. AP Professional, Cambridge, MA (1994)

30. Argyle, M., Cook, M.: Gaze and Mutual gaze. Cambridge University Press (1976)

31. Collier, G.: Emotional Expression. Lawrence Erlbaum Associates (1985)

32. T., B.R., G., C.J.: Children's understanding of emotional meaning in expressive body movement. Biennial Meeting of the Society for Research in Child Development (1996)

33. S., A., J.-C., M., Devillers, L.: A corpus-based approach for the modeling of multimodal emotional behaviors for the specification of embodied agents. In: HCI International, Las Vegas, USA (2005)

34. Dahl, S., Friberg, A.: Expressiveness of musician's body movements in performances on marimba. In Camurri, A., Volpe, G., eds.: Gesture-Based Communication in Human-Computer Interaction, 5th International Gesture Workshop, GW 2003, Genova, Italy, April 2003, Selected Revised Papers. Volume LNAI 2915., Berlin Heidelberg, Springer-Verlag (2004) 479–486

35. Dahl, S., Friberg, A.: Visual perception of expressiveness in musicians' body movements. (submitted)

36. Whissel, C.: The dictionary of affect in language. Emotion Theory, Research and Experience **4** (1989)

37. Mancini, M., Hartmann, B., Pelachaud, C., Raouzaiou, A., Karpouzis, K.: Expressive avatars in MPEG-4. In: IEEE International Conference on Multimedia & Expo, Amsterdam (2005)

Detecting Emotional Content from the Motion of an Orchestra Conductor

Tommi Ilmonen and Tapio Takala

Helsinki University of Technology,
Telecommunications Software and Multimedia Laboratory
Firstname.Lastname@tml.hut.fi

Abstract. In this paper we present methods for analysis of the emotional content of human movement. We have studied orchestra conductor's movements that portrayed different emotional states. Using signal processing tools and artificial neural networks we were able to determine the emotional state intended by the conductor. The test set included various musical contexts with different tempos, dynamics and nuances in the data set. Some context changes do not disturb the system while other changes cause severe performance losses. The system demonstrates that for some conductors the intended emotional content of these movements can be detected with our methods.

1 Introduction and Background

Emotionally aware computer systems are a new and interesting research field. Emotion detection is a crucial part of making a system that can take emotions into account. Especially facial expression and voice have been tackled by researchers. These modalities have also been combined for more robust gesture recognition [1]. Even mice have been made emotion-aware. Physiological signals can also be used to track the emotional state of a person [2]. Kang has published a system that tracks the emotional state of a video stream [3].

In contrast little research has been published that would use hand or body motion as the starting point. Drosopoulos has published a system that aims to detect emotions based on the gestures that a person makes [4].

Movements usually convey more than emotions, there is a context for them. A unique feature of our research is that context changes were included in the tests. In these tests the context changes were presented by change of musical parameters; dynamics, tempo and character.

2 Collecting and Processing Data

Emotions cannot be measured directly. In these tests we asked a person to manifest a feeling. Conducting brings another layer of expression parallel with the emotional content — musical content and context. Nuances (staccato, legato)

and dynamics (piano, forte) also affect the motion. The musical environment is the context of the motion.

The conductors wore a data suit that contained magnetic motion tracking sensors. The conductor was asked to conduct short a passage of music without a band with given emotional content. The emotional and musical content changed during the passage. Musical parameters were also varied and expressed simultaneously, resulting in superposition of parameters. Variables were dynamics (piano / forte) and character (staccato / no character / legato). Passages were performed in four tempos (about 56, 80, 120 and 160 beats per minute). Since the emotions also varied the number of permutations of all parameters gets easily very large. The data set used in analysis contains about 20 takes per conductor with each take containing four combinations of parameters. Since we want a separate test set a few extra takes were also recorded — duplicating some of the parameter combinations of the first data set.

The process by which humans detect the emotional content of motion is not known. At any rate the form of the motion must play a role in the recognition process. To somehow mimic this process we calculate parameters (features) from the motion. This calculation acts as preprocessing for ANNs. The preprocessing must transform the absolute motion curves to more general and abstract parameters. These parameters should not be affected by tempo, nuances or dynamics. We tested various preprocessing methods. We used the Cartesian coordinate position and rotation metrics as the basic information. These parameters were then transformed to velocity, acceleration and curvature spectrograms, histograms and filterbank outputs. We used artificial neural networks (ANNs) as analysis tool. To produce the figures in this paper, only self-organizeng maps (SOM) were used.

3 Results

In general, we found that the system can detect emotions implied by hand movements. The performance of the system depends heavily on the conductor. In these tests only two conductors participated.Results in this paper were obtained with the conductor that was easier for the system to analyze.

It was found that the characters legato and staccato confused the ANNs greatly. Since we found no way to fix the problem these characteristics were left totally out of the analysis. As a result the parameters that were varied in the test set were dynamics, tempo and emotion. We were only interested in emotion — the other parameters were varied to represent changing context. The effect of tempo range used was briefly tested. When the system only needs to analyze tempo 80 beats per minute it performs much better than when given the full range of tempos (see figure 1).

A widely used analysis approach is to classify the emotions to a fixed number of emotions e.g. categorization of emotions to N mutually exclusive classes. In this case we used neural networks that had N outputs — one output per emotion. The output with largest activation is then assumed to represent the emotion. Based on this one can calculate the confusion matrix that indicates how well
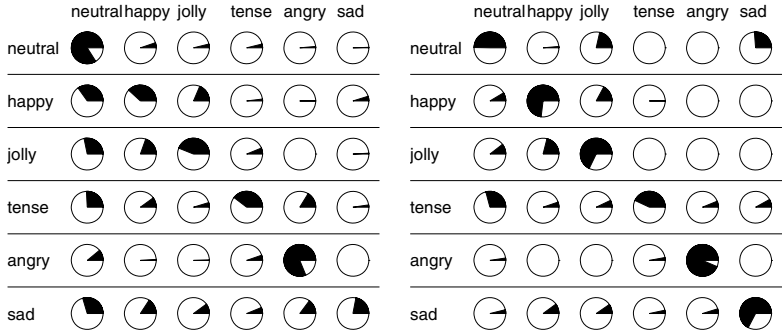
(a) The confusion matrix when a wide range of tempos is used

(b) The confusion matrix when the tempo is fixed to 80 beats per minute

Fig. 1. Confusion matrices in various situations

the analysis worked (figure 1(a)). In the matrix the title of each row indicates the intended emotion and the pie charts on the row indicate how the system interpreted the motion samples corresponding to that emotion. In the ideal case the diagonal elements of the matrix would be one and all others zero. By using random choice, one would get 17 percent (one out of six) of choices correct. In figure 1(a) the ratio of correct choices is between 20 and 80%.

By considering the emotional space as a low-dimensional continuum one can drop the number of dimensions of the emotional space. We used two-dimensional space illustrated in figure 2. The same figure shows how emotions we used were mapped to the space. We chose locations that appeared feasible. The ANN was trained to create an output vector that is correctly located in this emotional space. Figure 3 displays results obtained with one ANN. In the optimal case the ANN output of each emotion would be clustered to match the positions given in figure 2.
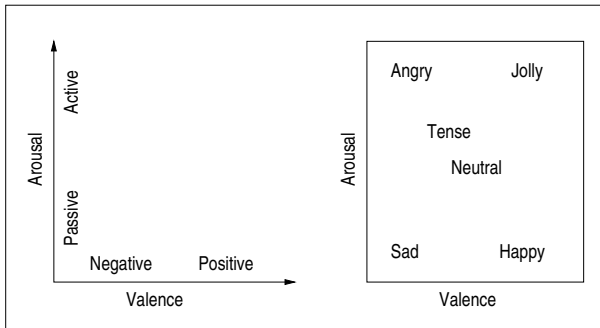

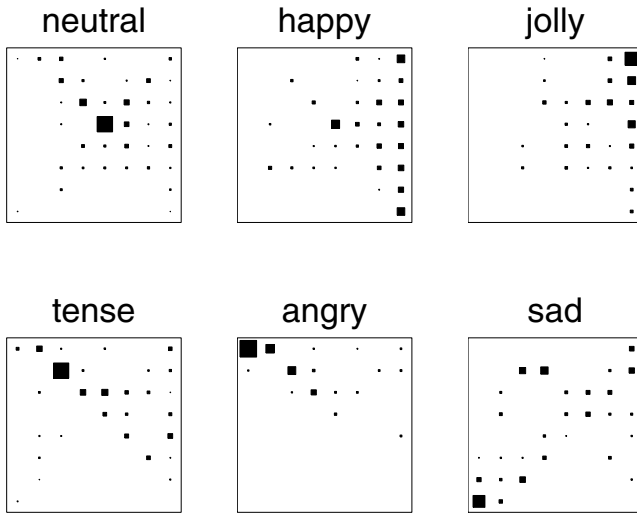**Fig. 2.** Definition of arousal/valence space and positions of emotions used in the space

neutral   happy   jolly

tense   angry   sad

Fig. 3. Distribution of ANN-estimations in arousal/valence space

4   Conclusions and Future Work

These are first results on a new field of study — how to determine the emotional content of human motion. As a qualitative result we can state that the gestural manifestation of emotions can be detected with computers. With only two test persons little can be said about how the system might work in the general case.

References

1. Chen, L., Tao, H., Huang, T., Miyasato, T., Nakatsu, R.: Emotion recognition from audiovisual information. In: IEEE Second Workshop on Multimedia Signal Processing. (1998) 83–88
2. Healey, J., Picard, R.: Digital processing of affective signals. In: Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing. Volume 6. (1998) 3749–3752
3. Kang, H.B.: Affective content detection using hmms. In: MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia, ACM Press (2003) 259–262
4. Drosopoulos, A., Balomenos, T., Ioannou, S., Karpouzis, K., Kollias, S.: Emotionally-rich man-machine interaction based on ges-ture analysis. In: Universal Access in HCI: Inclusive Design in the Information Society, Crete, Greece (2003) 1372–1376

Some Experiments in the Gestural Control of Synthesized Sonic Textures

Daniel Arfib[1], Jean-Michel Couturier[1], and Jehan-Julien Filatriau[1,2]

[1] CNRS-LMA, 31 chemin Joseph Aiguier,
13402 Marseille Cedex 20, France
{arfib, couturier}@lma.cnrs-mrs.fr
[2] Laboratoire de Télécommunications et Télédétection,
Université Catholique de Louvain (UCL), Belgium
filatriau@tele.ucl.ac.be

Abstract. In this paper, we introduce some exploratory ideas and applications involving the gestural control of sonic textures. Three examples of how the gestural control of synthesized textures can be implemented are presented: scratching textures, based on the gesturalized exploration of a visual space; dynamic noise filtering, where gestures influence a virtual slowly moving string used to filter a noise; and breathing textures, where the metaphor of breathing is used in the sound as well as in the gestural control. Lastly, we discuss how to find connexions between appropriate gestures and sonic texture processes, with a view to producing coherent and expressive digital musical instruments.

1 Sonic Textures and Gestural Control

Sonic textures are characterized by both microscopic and macroscopic features: on the short term scale, they are composed of a series of microstructural components which are subject to some randomness; whereas on the long term scale, some temporal and spectral cohesion is preserved. Sonic textures can result either from a process of computer analysis and synthesis or from synthesis alone [3]; the three examples we present in this paper are of the latter kind. Sound synthesis can be controlled by gestures using sensors connected to a computer. The link between gestural data and the synthesis parameters is called "*Mapping*" [1]. It is only when the gesture is properly linked to the sonic process, not only technically but especially at the human emotional level, that one can speak of a digital musical instrument.

2 Gestural Control of Sonic Textures: Some Examples

2.1 Scratching Textures

This digital instrument prototype is based on the gesturalized exploration of a visual space. It involves the real-time implementation of the Functional Iteration Synthesis (FIS) [3] driven by a bi-manual gestural control using a tablet-screen

Fig. 1. Left: the scratched gestural control using a tablet and a joystick. Right, two orbits created by direct control (orbit 1) and by parametric control (orbit 2).

and a joystick. FIS is an algorithm resulting from the wave terrain synthesis [6], where the terrains are obtained by iterating non-linear functions. An orbit is then traced on the three-dimensional surface to generate a waveform corresponding to the variations in the elevation of the trajectory over the terrain.

A gestural control inspired from the surface scratching allegory was developed [7]. The exploration of the wave terrains is carried out here either by performing linear trajectories (using the direct mode) or looping trajectories (using the parametric mode) (Fig. 1). In the direct mode, the orbit corresponds to the actual trajectory drawn by the user on the tablet screen, and the spectral features of the texture depend directly on the performers hand movements. In the parametric mode, trajectories are generated via three control parameters: the center of the trajectory drawn by the pen on the tablet, and its radius and its velocity, both of which can be modulated via a joystick. This parametric control makes it possible to overcome the limitations inherent to direct control and to create pseudo-pitched sonic textures.

2.2   Dynamic Noise Filtering

In the Filtering String instrument, gestures are used to control a virtual slowly moving string (based on a spring-mass model), which is used to filter a noise [2]. The string shape drives the gains of 32 filters and is displayed on a screen; the string model controls both the sound and the graphics. The choice of resonant frequencies and the quality factors (linked to the bandwith) of the filters give the sound a basic color; the motion of the string adds fluctuations to the sound. With this technique, it is possible to create a texture with complex but natural variations, because the physical behaviour of the string is well known. The user acts on the sound fluctuations indirectly, by interacting with the dynamic string using a graphic tablet and a multi-touch surface (Fig. 2 left). The position and pressure of the stylet on the graphic tablet affect the sonic texture by changing the intrinsic properties of the string (such as its tension, stiffness, and damping).

Fig. 2. Left: the gestural control of a filtering string. Right: an alternating gesture linked to breathing textures.

Forces can be applied to the virtual string by exerting pressure on the multi-touch surface; this changes the strings equilibrium position and thus modifies the frequency spectrum of the texture. With one hand (using the stylet), the user configures the dynamic system and thus determines how the string will respond to the other hand gestures (on the touch surface). The effects of the touch pad gestures on the sound will depend on the string configuration. For example, at low stiffness values, the string will move slowly and will not respond to fast movements on the touch pad: in this case, fast gestures are filtered by the dynamic string. With his hands, the user can impart energy on the string and change the way it moves; the sonic texture will evolve, although it keeps its identity.

2.3   Breathing Textures

Playing this instrument requires making ecological hand gestures with an alternating pattern, which interact with a breathing sound production process. Windy textures can be produced by multiplying a band-limited noise signal by a sine wave (amplitude modulation) [5]. In this way, the band-limited spectrum is translated by the sine wave frequency value, giving rise to the perceptual sensation of a definite pitch, while the timbre is noisy but quite smooth. The control parameters are the perceptual pitch, the noisiness, and the amplitude of the sound. The metaphor of breathing (inhaling and exhaling) can be used, for example, by alternating two different sonic textures, which can also evolve with time. The most natural way of gesturalizing this metaphor is to use bimanual gestures, with the two hands moving in opposite directions (Fig. 2 right). Generally speaking, it can be appropriate to use familiar gestures (which are sometimes called ecological gestures) mimicking common manual activities and to link them to these sounds.

3   Textures and Gestures

Sonic textures are specific: they are perceived as sound masses with a rather indefinite pitch, and the usual attack part is often replaced by a series of transients

initiated throughout the duration of the sound. Another aspect of sonic textures links up with the fact that many natural sounds are textures resulting from the movements of bodies, as well as those of fluids and gases. It is therefore possible to set up a relationship between the energy of the performers movements and the evolution of the sound (see an earlier study by Hunt [4]). The best way of determining which gestures should be used to control a texture is to create a mental image of the texture before trying to find the most appropriate gestures for controlling it, rather than looking first for the most efficient artificial links.

4   Conclusion

We have established that it is possible to create new digital instruments using sonic textures, taking their temporal and spectral specificities into account. We have concluded from our experiments that the gestures associated with the production of sonic textures should correspond to the ecological nature of these sounds, and that the gestural control is at least as important as the synthesis algorithm. We intend in the future to investigate the most suitable ecological gestures for producing ecological sounds, paying due attention to the specificities of sonic textures. Potential applications such as the musical possibilities of these instruments or the emotional textural rendering will also be investigated.

References

1. Arfib, D., Couturier, J.M., Kessous, L.: Design and Use of Some Digital Musical Instruments, in A. Camurri & G. Volpe (Ed) Gesture-Based Communications in Human-Computer Interaction, Lectures notes in Artificial Intelligence, LNAI 2915, pp. 509-518. Springer Verlag, 2004.
2. Arfib, D., Couturier, J.M., Kessous, L.: Gestural Strategies for specific filtering processes, in Proceedings of Digital Audio Effects 2002 conference (DAFx02), pp. 1-6, Hamburg, Germany, 26-28 sept 2002.
3. Di Scipio, A.: Synthesis of environmental sound textures by iterated non linear functions and its ecological relevance to perceptual modelling, Journal of New Music Research, Vol. 31 p. 109-117, 2002.
4. Hunt, A., Kirk, R.: Mapping Strategies for Musical Performance, Trends in Gestural Control of Music, CD-rom, Edited by M. Wanderley and M. Battier, Ircam, 2000.
5. Mathews, M.: The Technology of Computer Music, MIT Press, Cambridge, MA, 1989.
6. Roads, C.: The Computer Music Tutorial, MIT Press, Cambridge, MA, 1996.
7. Wessel, D., Wright, M., Schott, J.: Intimate Musical Control of Computers with a Variety of Controllers and Gesture Mapping Metaphors, in Proceedings of the New Instruments for Musical Expression conference (NIME02), Dublin, Ireland, May 24-26, 2002.

Deixis: How to Determine Demonstrated Objects Using a Pointing Cone

Alfred Kranstedt, Andy Lücking, Thies Pfeiffer, Hannes Rieser, and Ipke Wachsmuth

Collaborative Research Centre SFB 360,
Faculty of Technology and Faculty of Linguistics,
University of Bielefeld, 33594 Bielefeld, Germany
`{akranste, tpfeiffe, ipke}@techfak.uni-bielefeld.de,`
`{andy.luecking, hannes.rieser}@uni-bielefeld.de`

Abstract. We present a collaborative approach towards a detailed understanding of the usage of pointing gestures accompanying referring expressions. This effort is undertaken in the context of human-machine interaction integrating empirical studies, theory of grammar and logics, and simulation techniques. In particular, we attempt to measure the precision of the focussed area of a pointing gesture, the so-called pointing cone. The pointing cone serves as a central concept in a formal account of multi-modal integration at the linguistic speech-gesture interface as well as in a computational model of processing multi-modal deictic expressions.

1 Introduction

Research in cognitive science shows that deixis, pointing, or demonstration is at the heart of reference. On the other side, the robust grounding of reference in situated human-machine communication is an open issue until now. In this paper we concentrate on pointing gestures in deictic expressions. Following McNeill [14], we distinguish between abstract pointings and pointings into concrete domains. Here we focus on pointings into concrete domains co-occurring with verbal expressions.

In our research on human computer interfaces for natural interaction in Virtual Reality (VR) we employ an anthropomorphic agent called Max who is able to produce synchronised output involving synthetic speech, facial display and hand gestures [8]. Doing so, we focus on scenarios in the construction task domain, where a kit consisting of generic parts is used to construct models of mechanical objects and devices. A typical setting consists of a human user instructing a VR system represented by Max in aggregating composite objects. Speech and gesture are used to specify tasks and select relevant referents. To improve the communicative abilities of Max, he will be equipped with a natural pointing behaviour meeting the requirements of deictic believability [12].

A central problem we are faced with is the vagueness of demonstration. The question is how to determine the focus of a pointing gesture. To deal with that, we establish in the course of a parameterisation of demonstration (Section 2) the concept of a pointing cone. For our ongoing empirical studies we developed novel empirical methods using tracking technology and VR simulations to collect and evaluate analytical data (Section 3). In Section 4 we describe in the context of a theoretically

motivated multi-modal linguistic interface how the empirically fixed pointing cone can be used to integrate the content of the demonstration, determined *via* this cone, with the content of the verbal expression. The application of the pointing cone concept in computational models for (1) reference resolution and (2) the generation of multi-modal referring expressions embedded in our agent Max is outlined in Section 5. Finally, in Section 6 we discuss the trade-offs of our approach.

2   The Parameters of Demonstration

In accordance with Kita [6] we conceive of pointing as a communicative body movement that directs the attention of its addressee to a certain direction, location, or object. In the following we concentrate on hand pointing with extended index finger into concrete domains. In the context of multimodal deictic expressions pointing or demonstration serves to indicate what the referent of the co-uttered verbal expression might be [5]. If we want to consider the multiple dimensions of this kind of deixis more systematically, then we must account for various aspects:

(a) Acts of demonstration have their own structural characteristics. Furthermore, co-occurrence of verbal expressions and demonstration is neatly organised, it harmonises with grammatical features. Gestural and verbal information differ in content. This results from different production procedures and the alignment of different sensory input channels. The interaction of the differing information can only be described via a multi-modal syntax-semantics interface.

(b) Beside the referential functions of pointing discussed in literature (see e.g. [6] and [5]), which draw on the relationship between gesture form and its function, we concentrate on two referential functions of pointing into concrete domains depending on the spatial relationship between demonstrating hand and referent. If an act of pointing uniquely singles out an object, it is said to have *object-pointing* function; if the gesture refer only with additional restricting material it is assigned *region-pointing* function. As shown in earlier studies [13], classifying referential functions needs clear-cut criteria for the function distinction.

(c) Pointing gestures are inherently imprecise, varying with the distance between pointing agent and referent. To determine the set of entities delimited by pointing, we have to analyse which parameters influence the topology of the spatial area singled out by the gesture. As a first approximation we can model this area as a cone representing the resolution of pointing. Empirical observations indicate that the concept of the pointing cone can be divided into two topologically different cones for object- and for region-pointing, with the former having a narrower angle than the latter.

(d) Pointing gestures and speech that constitute a multi-modal utterance are time-shared. One point of interest, then, is whether there is a constant relationship in time between the verbal and the gestural channel. Our investigation of temporal *intra*-move relations is motivated by the synchrony rules stated in [14]. Since the so-called "stroke" is the meaningful phase of a gesture, from a semantic point of view the synchronisation of the pointing stroke and its affiliated speech matters most.

(e) With respect to dialogue, a further point of interest is whether pointings affect discourse structure. To assess those *inter*-move relations, the coordination of the

gesture phases of the dialogue participants in successive turns has to be analysed. For instance, there is a tight coupling of the retraction phase of one agent and the subsequent preparation phase of the other suggesting that the retraction phases may contribute to a turn-taking signal.

To sum up, elaborating a theory of demonstration means at least dealing with the following issues: (a) the multi-modal integration of expression content and demonstration content, (b) assigning referential functions to pointing, (c) the pointing region singled out by a demonstration ("pointing cone"), (d) *intra*-move synchronisation, and (e) *inter*-move synchronisation.

3   Empirical Studies on the Pointing Cone

To address the issues named in the preceding section we started to conduct several empirical studies in a setting where two subjects engaged in simple object identification games. One subject has the role of the "description-giver". She has to choose freely among the parts of a toy airplane lying on a table equally distributed, the pointing domain (Fig. 1a), and to refer to them. The other subject, in the role of the "object-identifier", has to resolve the description-givers reference act and to give feedback. Thus, reference has to be negotiated and established using a special kind of dialogue game.

In a first study described in [13] the object identification games were recorded using two digital cameras, each capturing a different view of the scene. The annotations of the video data comprise speech, gesture phases, and the structure of the dialogue games in terms of dialogue moves. This study yields useful results concerning the temporal relations of pointing and speech both within a single dialogue move and between the moves of the dialogue participants. However, concerning the topology of the pointing cone no reliable results could be obtained based only the recorded video data.

3.1   Tracker-Based Data Recording

To obtain more exact data concerning the pointing behaviour we use a marker-based optical tracking system for the body of the description-giver and data gloves for the fine-grained hand postures. The optical tracking system uses eight infrared cameras arranged in a cube around the setting to track optical markers each with a unique 3-dimensional configuration. A software module integrates the gathered information providing absolute coordinates and orientations. We track head and back of the description-giver to serve as reference points. The arms are tracked by two markers each, one for the elbow and one for the back of the hand. The hands are tracked using CyberGloves® measuring flexion and abduction of the fingers directly.

The information provided by both tracking systems (Fig. 1a) is integrated in a graph-based geometrical model of the user's posture (Fig. 1b). This is done in real-time using the VR frameworks Avango [18] and PrOSA [11]. Special recording modules are attached to the geometric user model to make the recorded data available for annotation and stochastic analysis (Fig. 1c).
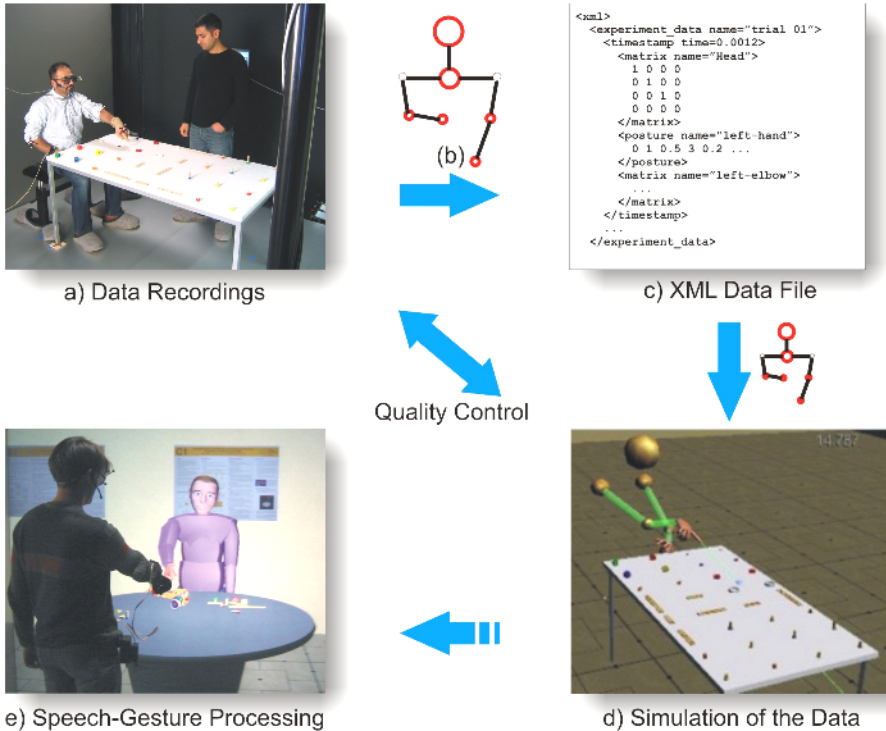
Fig. 1. The description-giver is tracked using optical markers and data gloves (a). The data is integrated in a geometrical user model (b) and written to an XML file (c). For simulation the data is fed back into the model and visualised using VR techniques (d). The findings are transferred to improve the speech-gesture processing capabilities of the agent Max (e).

To test the experimental setting we ran a preliminary study in November 2004 in which our primary concerns were the question of data reliability and the development of methods for analysing the data. The following section describes a simulative approach to support raters with visualisations of the collected data.

3.2 Simulation-Based Data Evaluation

For the simulation we use VR techniques to feed the gathered tracking data (Fig. 1c) back into the geometric user model, forming now the basis of a graphical simulation of the experiment (Fig. 1d). This simulation is run in a CAVE-like environment, where the human rater is able to walk freely and inspect the gestures from every possible perspective. While doing so, the simulation can be run back and forth in time and thus, e.g., the exact time-spans of the strokes (interval of maximal extension) can be collected. To further assist the rater, additional features can be visualised, e.g., the pointing beam or its intersection with the table. For the visualisation of the subject we use a simple graphical model (Fig. 1d) providing only relevant information.

For a location independent annotation we created a desktop-based visualisation system where the rater can move a virtual camera into every perspective possible and generate videos to facilitate the rating and annotation process when the graphic machines for the real-time rendering are not available. Using the annotation software, these videos can be shown side-a-side in sync with the real videos and provide additional perspectives, e.g., seeing through the eyes of the instruction-giver.

3.3 Computation of Pointing Beam and Pointing Cone

The principal aim of collecting analytical data was to fix the topology of the pointing cone and to measure its size.

A pointing beam is defined by its origin and its direction, the pointing cone in addition by its apex angle. Therefore, to grasp the spatial constraints of pointing, one has to identify the anatomical anchoring of origin and direction in the demonstrating hand and to calculate the apex angle of the pointing cone.

There are four different anatomical parts (the three phalanxes of the index finger and the back of the hand) at disposition for anchoring. To discriminate between them a hypothetical pointing beam is generated for each of them (Fig. 2a). We will choose the anchoring resulting in the least mean orthogonal distance over all successful demonstrations between the hypothetical pointing beam and the respective referent.
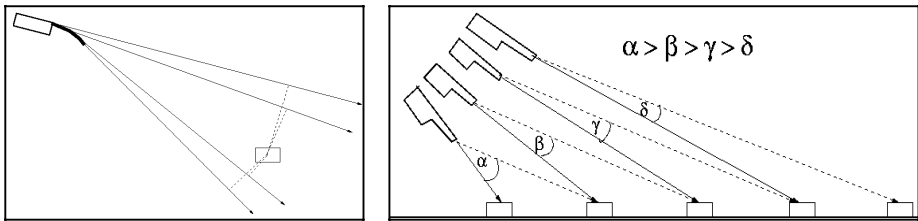

Fig. 2. a) Four hypothetical pointing beams anchored in different anatomical parts of the hand; b) The angles between the beams to the referent and their next neighbours decrease with the increasing distance to the referent. The dashed arrows represent the beams to the next neighbour.

Given the anchoring thus obtained, the calculation of the apex angle of the pointing cone can be done as follows: For each recorded demonstration the differing angle between the pointing beam and a beam with the same origin but directed to the nearest neighbour has to be computed. The computed angles decrease with the increasing distance between the demonstrating hand and the referent analogously to the perceived decreasing distance between the objects, see Fig. 2b.

We pursue two strategies for the calculation of the apex angle. In one experimental setting the description-givers are allowed to use both, speech and gesture, to indicate the referent. Analysing this data, we have to search for the differing angle correlating with a shift to more discriminating verbally descriptions. This angle indicates the borderline of the resolution of pointing the description-givers manifests. In the other experimental setting the description-givers are bounded to gestures only. In this data we have to search for the differing angle correlating with the distance where the number of failing references exceeds the number of successful references. This angle indicates

the borderline in the object density where the object-identifier cannot identify the referent by pointing alone.

We assume that these two borderlines will be nearly the same, with the former being a little bit broader than the latter due to the demonstrating agent's intention to ensure that the addressee is able to resolve the reference act. The corresponding angles define the half apex angle of the pointing cone of object-pointing.

A first assessment of the apex angle of this pointing cone using a similar calculation based on the video data recorded in our first studies resulted in a half apex angle between 6 and 12 degrees, see [9]. However, these results can be only taken as a rough indication.

To establish the apex angle of the pointing cone of region-pointing we have to investigate the complex demonstrations including verbal expressions referring to objects in the distal region. We hope that we can determine the contrast set from which the referent is distinguished by analysing the attributes the description-giver uses to generate the definite description. The location of the objects in the contrast set gives a first impression of the region covered by region-pointing.

In the next section, we introduce a formal attempt to integrate gestural deixis, in particular the pointing stroke, in linguistic descriptions, aiming at a theoretical model of deixis in reference [17].

4   A Multi-modal Linguistic Interface

4.1   Complex Demonstrations: Object and Restrictor Demonstration

Objects originating from pointing plus definite descriptions are called complex demonstrations ("CDs"). The pointing stroke is represented as "↘" indicating the start of the stroke in the signal and hence its scope. (1) presents a well-formed CD "↘this/that yellow bolt" embedded into a directive as against (1') which we consider as being non-well-formed in that the pointing gesture the demonstrative selects for is missing.

(1) Grasp ↘this/that yellow bolt. (1') *Grasp this/that yellow bolt.

A unified account of CDs will opt for a compositional semantics to capture the information coming from the verbal and the visual channel. CDs are considered as definite descriptions to which demonstrations add content either by specifying an object independently of the definite description or by narrowing down the description's restrictor. We call the first use "object demonstration" and the second one "restrictor demonstration".

Working on this assumption, demonstrations (a) act like verbal elements in providing content, (b) interact with verbal elements in a compositional way, (c) may exhibit forward or backward dynamics depending on the position of ↘.

4.2   Interpretation of CDs

The central problem is how to interpret demonstrations. This question is different from the one concerning the ↘'s function tied to its position in the string. We base the discussion on the following examples showing different empirically found ↘ positions and turn first to "object demonstration":

(2) Grasp ↘ this/that yellow bolt.   (3) Grasp this/that ↘yellow bolt.

(4) Grasp this/that yellow ↘bolt.   (5) Grasp this/that yellow bolt↘.

Our initial representation for the speech-act frame of the demonstration-free expression is

(6) $\lambda N \; \lambda u(N \; \lambda v \; F_{dir} \; (grasp(u, v)))$.

Here "F_{dir}" indicates directive illocutionary force; "N" abstracts over the semantics of the object-NP/definite description, and "(grasp(u, v))" presents the proposition commanded. The ↘ provides new information. If the ↘ is independent from the reference of the definite description the only way to express that is by extending (6) with "v = y", where y is a variable introduced by the pointing gesture:

(7) $\lambda N \; \lambda u \; \lambda y(N \; \lambda v \; F_{dir} \; (grasp(u, v) \; \wedge \; (v = y)))$.

The idea tied to (7) is that the reference of *v* and the reference of *y* must be identical, regardless of the way in which it is given. Intuitively, the reference of *v* is given by the definite description "$\iota z(yellowbolt(z))$" and the reference of *y* by ↘. The values of both information contents are independent of each other.

On the other hand, in the restrictor demonstration case the ↘ contributes a new property narrowing down the linguistically expressed one. The bracketing we assume for (3) in this case is roughly

(8) [[grasp] [this/that [↘yellow bolt]]].

To capture the restriction function, the format of the description must change. This job can be easily done by (9):

(9) $\lambda R \lambda W \lambda K.K(\iota z(W(z) \; \wedge \; R(z)))$

Here, K abstracts over the semantics of the directive, W is the predicative delivered by the noun, and R is the additional restrictor. The demonstration ↘ in (3) will then be represented simply by

(10) $\lambda y(y \in D)$,

where D intuitively indicates the demonstrated subset of the domain as given by the pointing cone. Under functional application this winds up to

(11) $\lambda K.K(\iota z(yellowbolt(z) \wedge z \in D))$.

Intuitively, (11), the completed description, then indicates "the demonstrated yellow bolt" or "the yellow-bolt-within-D".

4.3 Multi-modal Meaning as an Interface of Verbal and Gestural Meaning

Even if we assume compositionality between gestural and verbal content, we must admit that the information integrated comes from different channels and that pointing is not verbal in itself, *i.e.* cannot be part of the linguistic grammar's lexicon. The representation problem for compositionality becomes clear, if we consider formula (12)

(12) $\lambda Q \lambda N \; \lambda u \; N(Q(\lambda y \; \lambda v F_{dir} \; (grasp(u, v) \; \wedge \; (v = y)))) \; \lambda P.P(a)$ /*[grasp + ↘]

Evidently, (12) does more than a transitive verb representation for "grasp" should do. It has an extra slot Q designed to absorb the additional object, *i.e.* the demonstration $\lambda P.P(a)$. We must regard (12) as a formula belonging to a truly multi-modal domain, where, however, the channel-specific properties have been abstracted away from. This solution only makes sense, however, if we maintain that demonstration contributes to the semantics of the definite description used.

This idea is illustrated in greater detail in Fig. 3. The interface construction shown there for (12) presupposes two things: The lexicon for the interface contains expressions where meanings of demonstrations can be plugged into; demonstrations have to be represented in the interface as well.
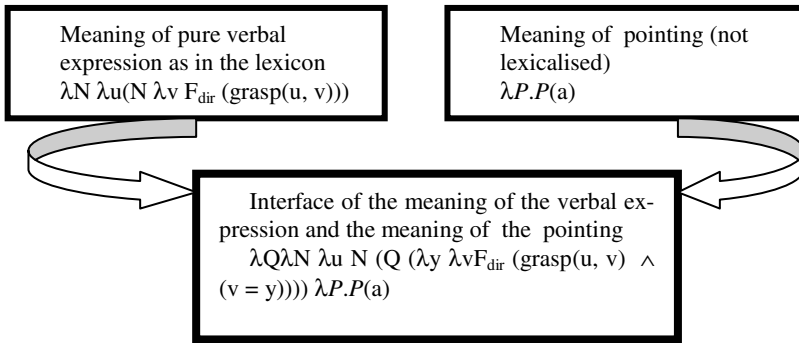
Fig. 3. Multi-modal interface: meanings from the verbal and the gestural channel integrated via translation of ↘

4.4 Underspecified Syntax and Semantics for Expressions Containing ↘

The varying position of ↘ can be captured in an underspecification model. The model coming nearest our descriptive interests is the *Logical Description Grammars* (LDGs).

A simplified graphical representation of inputs (1) and (3) is given in Fig. 4. '+' and '−' indicate components which can substitute ('+') or need to be substituted ('−'). Models for the descriptions in Fig. 4 are derived pairing off + and – -nodes in a one-to-one fashion and identifying the nodes thus paired. Words can come with several lexicalisations as can ↘-s.

The *logical description of the input* has to provide the linear precedence regularities for our example "Grasp the yellow bolt!" The *description of the input* must fix the underspecification range of the ↘. It has to come after the imperative verb. The *lexical descriptions for words* will also have to contain the type-logical formulas for compositional semantics as specified in (7) or (9).

Based on the syntax given in Fig. 4 and the type-logical formulas for compositional semantics specified in (12), we can now provide an interpretation for the speech act represented in

(13) F_{dir} (grasp(you, ιz(yellowbolt(z)))) $\wedge$ ιz(yellowbolt(z)) = a).

A full interpretation of (13) has to specify its having been performed, its success, the commitments it expresses and its satisfaction in the context of utterance *i*.
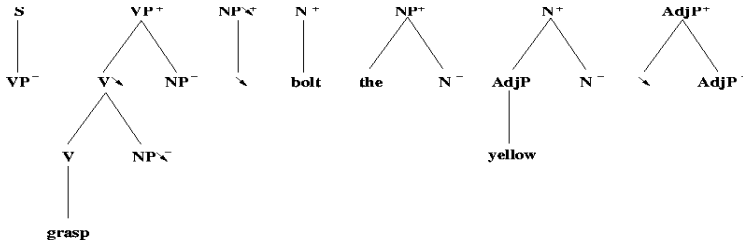

Fig. 4. Graphical representation of an input example in LDG

5 The Pointing Cone in Speech-Gesture Processing

In this section we discuss the relevance of pointing and pointing cones from the human-computer interaction perspective. The first part highlights the computational advantages for reference resolution from the view of speech and gesture understanding. In the second part it is demonstrated how the cone can be used on the production side to decide whether object- or region-pointing is appropriate for a specific deictic referring expression and how it influences content selection.

5.1 Reference Resolution

In our framework for interaction in VR complex demonstrations (CDs) are interpreted by the Reference Resolution Engine (RRE, [15]). Incoming CDs are parsed into sets of constraints over the current world model. The constraint satisfaction kernel accesses several heterogeneous knowledge bases (KB) for symbolic information such as type, colour or function and for geometrical information to generate solutions for each set of constraints. It uses fuzzy logic for a robust interpretation of symbolic categories (e.g. "red", "left of").

A simple set of constraints representing "the ↘yellow bolt" could look like this:

```
(inst ?x OBJECT)       (pointed-to instruction-giver ?x time-1)
                       (has-colour ?x YELLOW time-1)
(inst ?y TYPE)         (is-a ?y BOLT time-2)
(has-type ?x ?y time-2)
```

The RRE solves these constraints and returns a list of possible interpretations ordered by likelihood. As our scenes are highly dynamic, special consideration has to be given to the time course of the interpretation. A time stamp (`time-1`, `time-2`) is attached to each word or gesture and carried through all processing steps to synchronise each input with the state of the world at that time. For this each knowledge base maintains a history of localised views, one for each arm and head of the participants, over the time course of the uttering of the CD.

Geometric constraints formulated verbally, e.g., by "to the left of the block" involve a switch from a discrete, symbolic domain to the continuous domain of geometry and may therefore be highly ambiguous. This fuzziness is increased even more when several geometric constraints are combined, e.g. "to the left and behind the block". In contrast, the pointing cone is a device which has a straight forward representation in the geometric domain. Entities pointed to are resolved directly with optimised intersection algorithms. Geometric constraints in the accompanying speech can then be computed on a highly restricted subset of the world model, reducing the described problems.

Using the concept of the pointing cone the RRE computes the geometrical context of a CD with less cost, and thereby faster, while yielding more precise results.

5.2 Generation of Deictic Expressions

While much work concerning the generation of verbal referring expressions has been published, work on the generation of multi-modal referring expressions is rare. Most approaches use idealised pointing in addition or instead of verbal referring expressions; see e.g. [2], [16], [1] and [12]. Only Krahmer and van der Sluis [7] account for vague pointing and distinguish the three types *precise*, *imprecise*, and *very imprecise* pointing.

We propose an approach, for details *cf* [10], which integrates an evaluation of the discriminating power of pointing using the concept of pointing cones with a content selection algorithm for definite descriptions founded on the incremental algorithm published by [4].

Based on our empirical observations, we use the pointing cone to define the focus of a planned pointing gesture and distinguish the two referential functions object-pointing and region-pointing discussed above. As a first step, disambiguation of the referent by object-pointing is checked. Doing so, a pointing cone with an apex angle of 12 degree anchored in an approximated hand-position and directed to the referent is generated. If only the intended referent is found inside this cone, we can refer by conducting object-pointing without an additional description of the object uttered verbally. If object-pointing does not yield a referent, region-pointing is used to focus the attention of the addressee to a certain area making the set of objects inside this area salient. This set of salient objects is determined by the pointing cone of region-pointing characterized by a wider apex angle than the cone of object-pointing. In our current implementation we chose heuristically the value 25 degrees.

The objects inside this cone have to be distinguished by additional properties. For determining them we use an adapted version of the incremental algorithm of Dale and Reiter [4], which exploits domain-specific knowledge about typical properties to achieve a determined sequence in property evaluation and to avoid backtracking. This approach computes in linear time and the results fit well with the empirical findings. In our construction domain typically the property hierarchy, type, colour, relative size related to form, is used. The algorithm is adapted as much as relational properties are considered.

The results of the content selection algorithm are represented by a list of attribute-value-pairs, which are fed into a surface realisation module generating a syntactically

correct noun phrase. This noun phrase is combined with a gesture specification and both are inserted into a surface description of a complete multi-modal utterance. Based on these descriptions, an utterance generator synthesizes continuous speech and gesture in a synchronised manner to be uttered by Max [8].

6   Conclusion

The collaborative research presented in this paper scrutinised the issue of pointing in complex demonstrations. This issue was approached from interlocked perspectives, spanning the complete cycle of speech-gesture processing.

A genuine effort has been started in collecting multi-resolutional empirical data on deictic reference ranging from the high levels of speech acts down to the details of finger movements. The analysis of data on complex descriptions led to the notion of *pointing cone* fusing the parameters relevant for the discriminating power of pointing. A detailed procedure has been worked out to assess the geometrical properties of the pointing cone using tracking technology for measuring the pointing behaviours of subjects. Based on the described methods, the results of the studies will ultimately allow the fixation of a set of parameters relevant for the computation of the pointing cone's size and form. Furthermore, the sophisticated simulation of the collected data enriches the traditional video-based annotation approach; a technique that can easily be transferred to other topics of investigation.

The empirically justified concept of pointing cone enables an integrative approach to object- and region-pointing as part of complex demonstrations in concrete dialogue situations. In result, complex demonstration, and pointing as part of it, can be modelled in a more natural manner than in previous approaches. In utterance generation the pointing cone covers the object(s) to be made salient to the addressee. These objects constitute the contrast set for content-selection in planning a definite description. This idea in turn is taken up by the reference resolution procedure where the area of the cone is used to narrow down the search space. Finally, as has been shown with the multi-modal linguistic interface, the concept of the pointing cone enters into formal definitions of performance, success, commitments and satisfaction of speech acts containing complex demonstrations in an utterance's context.

Acknowledgement

References

1. André, E., Rist, T., and Müller, J.: Employing AI Methods to Control the Behavior of Animated Interface Agents. *Applied Artificial Intelligence*, 13:415-448, (1999).
2. Claassen, W.: Generating Referring Expressions in a Multimodal Environment. In Dale, R. et al. (eds.): *Aspects of Automated Natural Language Generation*. Springer, pp. 247-262, (1992).

3. Dale, R.: Cooking up Referring Expressions. In *Proceedings of the 27th Annual Meeting of the ACL.* Vancouver, pp. 68-75, (1989).

4. Dale, R. and Reiter, E.: Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science*, 18:233-263, (1995).

5. Kendon, A.: *Gesture: Visible Action as Utterance.* Cambridge University Press, (2004).

6. Kita, S (ed.).: *Pointing. Where Language, Culture, and Cognition Meet.* Lawrence Erlbaum Associates, Mahwah NJ, (2002).

7. Krahmer, E. and van der Sluis, I.: A New Model for the Generation of Multimodal Referring Expressions. In *Proceedings European Workshop on Natural Language Generation (ENLG 2003).* Budapest, pp. 47-54, (2003).

8. Kopp, S., and Wachsmuth, I.: Synthesizing Multimodal Utterances for Conversational Agents. *Comp. Anim. Virtual Worlds,* 15:39-52, (2004).

9. Kranstedt, A., Kühnlein, P., and Wachsmuth, I.: Deixis in Multimodal Human Computer Interaction: An Interdisciplinary Approach. In Camurri, A., and Volpe, G. (eds.): *Gesture-based Communication in Human-Computer Interaction*. Springer, LNAI 2915, pp. 112-123, (2004).

10. Kranstedt, A. and Wachsmuth, I.: Incremental Generation of Multimodal Deixis Referring to Objects. In *Proceedings European Workshop on Natural Language Generation (ENLG2005)*. Aberdeen, UK, pp. 75-82, (2005).

11. Latoschik, M. E.: A General Framework for Multimodal Interaction in Virtual Reality Systems: PrOSA. In *Proceedings of the Workshop The Future of VR and AR Interfaces - Multimodal, Humanoid, Adaptive and Intelligent.* IEEE Virtual Reality 2001, Yokohama, pp. 21-25, (2001).

12. Lester, J., Voerman, J., Towns, S., and Callaway, C.: Deictic Believability: Coordinating Gesture, Locomotion, and Speech in Lifelike Pedagogical Agents. *Applied Artificial Intelligence*, 13(4-5):383-414, (1999).

13. Lücking, A., Rieser, H., and Stegmann, J.: Statistical Support for the Study of Structures in Multimodal Dialogue: Inter-rater Agreement and Synchronisation. In *Proceedings of the 8th Workshop on the Semantics and Pragmatics of Dialogue (Catalog '04).* Univ. Pompeu Fabra, Barcelona, pp. 56-64, (2004).

14. McNeill, D.: *Hand and Mind: What Gestures Reveal about Thought*. University of Chicago Press, (1992).

15. Pfeiffer, T., and Latoschik, M.E.: Resolving Object References in Multimodal Dialogues for Immersive Virtual Environments. In *Proceedings of the IEEE Virtual Reality 2004*. Chicago, pp. 35-42, (2004).

16. Reithinger, N.: The Performance of an Incremental Generation Component for Multimodal Dialog Contributions. In Dale, R. et al. (eds.), *Aspects of Automated Natural Language Generation*. Springer, pp. 263-276, (1992).

17. Rieser, H.: Pointing in Dialogue. In *Catalog '04*. Op. cit., pp. 93-101, (2004).

18. Tramberend, H.: Avocado: A Distributed Virtual Reality Framework. In *Proceedings of IEEE Virtual Reality 1999*, pp. 14-21, (1999).

AcouMotion – An Interactive Sonification System for Acoustic Motion Control

Thomas Hermann[1], Oliver Höner[2], and Helge Ritter[1]

[1] Neuroinformatics Group,
Faculty of Technology, Bielefeld University, D-33501 Bielefeld, Germany
[2] Institute of Sport Science, University of Mainz, D-55099 Mainz, Germany

Abstract. This paper introduces *AcouMotion* as a new hard-/software system for combining human body motion, tangible interfaces and sonification to a closed-loop human computer interface that allows non-visual motor control by using sonification (non-speech auditory displays) as major feedback channel. AcouMotion's main components are (*i*) a sensor device for measuring motion parameters (*ii*) a computer simulation to represent the dynamical evolution of a model world, and (*iii*) a sonification engine which generates an auditory representation of objects and any interactions in the model world. The intended applications of AcouMotion range from new kinds of sport games that can be played without visual displays and therefore may be particularly interesting for people with visual impairment to further applications in data mining, physiotherapy and cognitive research. The first application of AcouMotion presented in this paper is *Blindminton*, a sport game similar to Badminton which is particularly adapted to the abilities of people with visual impairment. We describe our current system and its state of development, and we present first sound examples for interactive sonification using an early prototype. Finally, we discuss some interesting research directions based on the fact that AcouMotion binds auditory stimuli and body motion, and thus can represent a counterpart to the Eye-tracker device that exploits the binding of visual stimuli and eye-movement in cognitive research.

1 Introduction

Auditory information plays an important role for directing and coordinating human activity [1]. Almost every human activity, like closing a door or putting down a cup on the table, every foot step and almost any physical contact is connected with an acoustic feedback. This provides us with a variety of information about certain details of the interactants, e.g. their material, stiffness, energy, texture. In addition to such interaction sounds there are environmental sounds that give us useful hints on another level: they direct our attention (e.g. to an approaching car, or a mobile phone), or increase our awareness (think of the symphony of sounds in a wood from bird songs to the wind in the leaves).

Maybe it is because we interpret and use these informative signals so routinely and completely effortless that auditory information was not sufficiently appreciated for a long time. This may be one reason why our culture developed in a

rather visual-centered way. Particularly interaction with computer technology is still very eye-oriented. Over the last decade, sonification started to offer alternative, auditory displays which aim at addressing our auditory skills for analyzing data, particularly high-dimensional data [2]. These techniques now become more and more interactive, and they enable the user to navigate for instance data under analysis while perceiving in real-time an auditory representation [3].

In search for more ecological (i.e. natural and intuitive) interactions with auditory data displays, soon other controllers and interfaces than the keyboard and mouse came into view. When interacting with the environment, we usually employ our hands and arms, which are both very versatile, and which offer multi-dimensional controls. A combination of body motion and sound is not only interesting from the perspective of sonification, e.g. in musical performance it allows to bind dance and musical performance together.

This paper presents the new system *AcouMotion* that provides a link between motor activity and auditory feedback through sonification. *AcouMotion* is a hard-/software system that consists of a tangible sensor device, a dynamic model implemented in a computer simulation and a sonification engine. Interactions (resp. actions) with the interface object are mapped to manipulations of objects in the dynamic model. Reactions in the model world are displayed by sonification as the only feedback modality.

AcouMotion offers various applications, and we give in this paper a sketch of the possibilities plus a more detailed description of our first implemented application: using AcouMotion, we develop a new sport game for users with visual impairment that we call *'Blindminton'*, an adapted version of Badminton. The paper provides an overview of interactive sonification in Sec. 2, followed by the presentation of the AcouMotion system in Sec. 3, the basis for Blindminton (see Sec. 4). In the end in Sec. 5, we discuss different research possibilities of AcouMotion in diverse disciplines, with a focus on applications in Sports Science, Data Mining and cognitive research.

2   Interactive Sonification

Sonification is the use of non-speech audio for the representation of information [2]. Auditory displays, opposed to visual displays are inherently dynamic so that the information is in principle offered in the flow of time. While we can navigate visual displays actively by directing the visual focus, it seems that we are almost incapable of doing the equivalent in sound, apart from perhaps to focus our auditory attention on certain aspects of a perceived sound (e.g. to listen to the clarinets in a piece of music). However, this is the point where interaction comes into play. Usually, our environment is silent in the absence of excitation, and we ourselves cause excitation by interacting with the world. Due to the invariance of this principle evolution has optimized the human perceptual apparatus to cope with such multi-modal closed interaction loops. Inclusion of interaction in sonification is therefore a plausible step to better fit our sensomotor skills to the use of auditory display systems. While interactive sonification

mainly addresses the issue of investigating data by using interactive navigation controls, we here suggest an interface that uses also the data measured by a sensor device as source of sonification, and so provide acoustic motion feedback.

There are various techniques of sonification like Earcons, Auditory Icons, or the more complex techniques Parameter Mapping and Model-based Sonification [2, 4]. **Parameter Mapping Sonification** is a frequently used strategy to transform data streams $\{\mathbf{x}_i\}_i$ to acoustic streams [5]. Usually a mapping function $\mathbf{y} = f(\mathbf{x}) = \sigma(\mathbf{A}\mathbf{x})$ is applied to compute the acoustic attributes vector $\mathbf{y}$, frequently using a linear transformation A and a nonlinear distortion function σ. The components of $\mathbf{y}$ are sound synthesis parameters like for instance frequencies, amplitude, modulation indices.

Model-Based Sonification. (MBS) involves a dynamic model to mediate between the data and the sound [4]. Instead of controlling a sound synthesis engine, the data determines the setup of a dynamic system whose temporal evolution is the only process that generates sound (i.e. the sonification). The main advantages compared to Parameter Mapping are that MBS supports a generic design, tightly integrates interaction, and automatically generates acoustic relations that are intuitively understood (like the more a system is excited, the louder it typically sounds).

3   AcouMotion

Applications of the system *AcouMotion* use a mix of the sonification techniques mentioned above, e.g. Auditory Icons for displaying discrete events, Parameter Mapping for analogous data display and, for instance, Model-based Sonification for more complex data representations through audio.

The core idea behind AcouMotion is to employ sonification to create a new channel of proprioception allowing to perceptually relate body motion to virtual objects in a virtual space whose properties can be designed to support a wide range of different applications. AcouMotion connects three system components to implement this idea: (*i*) a *tangible sensor device* providing motion-related information, (*ii*) a *computer simulation model* formalizing the coupling between body motion (reflected in the sensory data provided by the tangible device) and the object dynamics in the virtual space, and (*iii*) a *sonification engine* for the perceptual rendering of the joint dynamics of body and modeled object states.

Body motion sonification in general bears the potential that gestural expressions (e.g. emotions) carry over to rhythmic and dynamic sound properties so that AcouMotion can be used for categorizing and monitoring gestural behavior. As bio-feedback system it allows the user to monitor his own activities on the background of a known 'auditory action template' and thus to evaluate differences in gesture execution which is interesting for motor learning and control in sports, but also for actors or choreography training.

From an application perspective, AcouMotion's underlying enhancement of proprioception for monitoring behavior in flexibly designable VR models of environments can also be used to analyze and support *training in sports*, to offer

novel ways of exploration and navigation in *interactive data mining*, to induce and stabilize therapeutic movement patterns in *physiotherapy*, and to offer new avenues for *investigating cognitive processes*.

In the following, we will illustrate only one research direction of our approach with a specific application example taken from the domain of *sport games* (further applications are pointed out in Sec. 5). In this example, the tangible controller will be a small handheld device, moved in a racket-like manner providing sensor signals allowing to drive a "virtual racket" in the model space. The model space will additionally contain a virtual game arena consisting of a floor, a demarcated field, a ball and reflecting walls. The sonification engine will create a real-time soundscape that allows to infer the distance, the position and the velocity of the virtual ball relative to the racket. Thereby, it will provide the player with non-visual proprioception how to perform a successful hit back of the ball to continue the game.

In this example the body motion is functional in the sense that only the "physical" contact of a virtual ball and virtual racket are relevant for playing the game. It is not determined how the player achieves this goal. Gestures, as a more indirect means of communication may be implemented in further game applications as AcouMotion is developed as a very general platform (e.g. a game "Gesture Imitation" where one blindfolded person challenges the opponent by performing a body gesture which results in a sonification, followed by the other player who may have 3 trials to reproduce the body gesture purely by trying to reproduce the sound).

3.1   Sensor Devices

In AcouMotion a variety of sensor devices can be used. They all have in common that they deliver real-time data about the user's physical activity. For a game like Blindminton, the position of one hand in high spatio-temporal resolution would be required. For other applications of AcouMotion, one might need only accelerations, or whole body movements, or force measurements.

As a general framework, we propose a *tangible sensor device* for AcouMotion. Users are familiar and highly skilled in using tools from everyday experience or many sport games, while purely gestural interactions are rare[1]. As a professional solution for our sensor device we use the Lukotronic motion capture system[2], which is able to track the 3D-position of a set of markers with a frame rate of 1200 Hz. The Lukotronic system consists of a set of fixed IR-cameras and flashing IR-markers. Using four markers mounted on a tangible device like a small racket allows us measuring the full 6D position/orientation of the racket. Furthermore, we can compute the velocities and accelerations at high accuracy from successive frames.

---

[1] However, the Theremin (see www.thereminworld.com/learn.asp) is a music instrument played via gestural interaction alone, and there are many motion forms like Tai-Chi or dance. Thus even cameras are suited sensors to be used in AcouMotion.
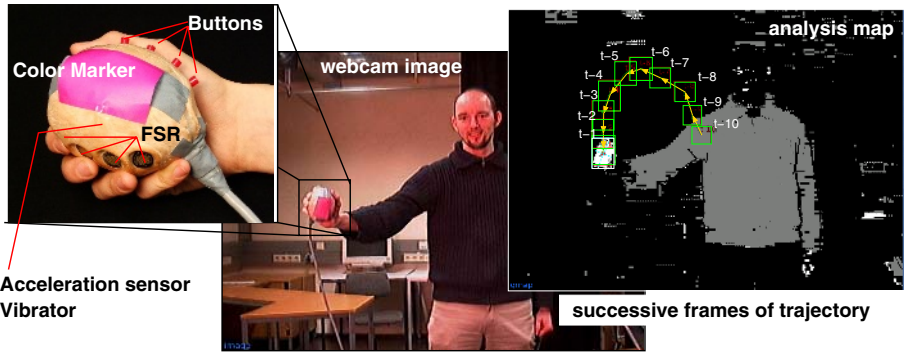
[2] http://www.lukotronik.com

Fig. 1. Webcam-based position estimates. The figure shows a player using the interface ball, and the webcam image including analysis results.

In addition, we search for a less expensive solution for AcouMotion. Our first experimental prototype is a haptic controller equipped with two 2D accelerometers and 5 force sensitive resistors to provide interactions like movement, rotation, shaking, squeezing, etc [6]. The interface is shown in Figure 1. A cable-free version using Bluetooth will free us from actual mobility limits and is a next target for development.

The frame rate is a crucial parameter in real-time interactions, and we currently obtain and process sensor data at 100 Hz, which is sufficiently high to create the illusion of latency-free control. However, the integration of accelerometer data to spatial coordinates is difficult and we require an independent means to eliminate drifts. Currently we solve this by a computer-vision based approach as shown in Fig. 1. By using a simple webcam, we are able to obtain 2D-coordinates of the ball at 25 Hz, and using a fixed sized visual marker, we can compute a rough estimate of the distance. We plan to fuse these estimates with predictions from the sensor data integration. Although the prototype of the haptic controller suffers from relatively low frame rates at this time, it is a valuable complement for the Lukotronic system in the Blindminton application. In combining the Lukotronic system and our haptic control device we are able to measure the position of the player's hand with high spatio-temporal solution and give also a feedback for crucial game events, in particular the hit of the ball. Using such additional haptic information augments the feeling of having control over the game. This is an essential condition for motivating flow-experiences during the game. Integrating haptic information is therefore believed as an important condition for people's motivation playing the game.

3.2   Computer Simulation

A dynamic model is the basis for representing processes and interactions in AcouMotion. The model represents the internal state of the AcouMotion system, and evolves according to its own "physical laws". In applications like the sport game like Blindminton, we suggest to use laws we are familiar with from

everyday experience (e.g. the ball flies through a 3-dimensional space influenced by gravitational force and aerodynamic resistance)[3].

To start we use a physical model in a 3D model space with a limited number of objects represented by their coordinates, velocities and orientations. For instance in a one-player version of Blindminton, the objects are the racket, the ball, and a set of planes and walls to model the game field and floor. In our current prototype, the ball is described by a sphere or radius r, mass m and its state vector $(\mathbf{x}_b[n], \mathbf{v}_b[n])$. In later versions it will also include the property of angular momentum. In a similar way, the racket is modeled by a cuboid. It is special in the way that its coordinate and orientation is strictly determined by the external sensor data. In contrast, the ball is free and only bound to follow the motion equations

$$m\ddot{\mathbf{x}} + R\dot{\mathbf{x}} + \nabla_x V(\mathbf{x}) = 0 \ \ \text{with} \ \ V(\mathbf{x}) = g\hat{z} \tag{1}$$

The state of the model is updated at a constant rate ΔT by using numerical integration. For instance, the ball is updated using

$$\mathbf{x}[n+1] = \mathbf{x}[n] + \mathbf{v}[n+1]\Delta T$$
$$\mathbf{v}[n+1] = \alpha\mathbf{v}[n] + \mathbf{a}[n+1]\Delta T$$
$$\mathbf{a}[n+1] = -\frac{1}{m}\nabla V(\mathbf{x}[n])$$

In addition to these update steps, the simulation needs to check at every time step whether there are interactions with objects (e.g. the ball and the virtual racket), and respond with update in this situation, like an elastic impact. Such event-based information is highly relevant for the auditory display.

While real-world settings have to operate with the existing physical laws, the computer simulation enables us to control any circumstances in principal, for instance the viscosity of the air. This might cause a retardation of the ball due to increasing aerodynamic resistance, etc. Thus, we can control the complexity and difficulty of the task in detail to create a challenging game.

3.3   Sonification Engine

Sonification bridges the gap between the only virtually existing model state and the auditory perception of the user. The sonification shall provide ample information to enable the user to operate whatever interactive activity is needed in the respective application. This could be the successful hitting of the ball, but also navigation in complex data spaces in more abstract settings.

Sound offers extensive possibilities to incorporate detailed information about ongoing processes. The sonification engine itself is an algorithm which receives as input the state of the model, and creates as output either the sound directly or control messages to a synthesis engine.

---

[3] However, this falls in the hand of an application designer. For instance this model can be a sonification model so that interactions may be used to analyze high-dimensional data as described in Sec 5.

Practically we use Supercollider [7], an object oriented language similar to SmallTalk for the implementation of the model, and the Supercollider sound server for the computation of audio data. The sonification engine can be exchanged easily by other implementations in C++ or our graphical simulation environment Neo [8] since communication to the sound server is achieved via Open Sound Control (OSC).

Sensor device, computer simulation model and sonification engine are connected via OSC interfaces, allowing easy exchange of sensors, or distribution on different computers.

Basis elements of our auditory display are (*i*) *continuous sound streams* which convey information by the change of acoustic attribute (an example is a pulsed sound whose pulse rate represents distance to the player). (*ii*) *discrete sound events*, which are used to communicate discrete event (e.g. physical contact interactions in the model) (*iii*) *ambient elements* like sound effects, that influence the overall display.

4   Blindminton – A Sonification-Based Sport Game for Adapted Physical Activity

In this section we focus on our first application of AcouMotion, a new sport game called *Blindminton*. Blindminton is providing a test case for several applications focusing on the excellent auditory perception skills which are highly adapted for people with visual impairment due to their enhanced everyday use. It is an application of the transdisciplinary method of interactive sonification in the interdisciplinary research field of *Adapted Physical Activity (APA)*, a relatively new focus within physical education and kinesiology for people with deficiencies, disabilities, handicaps or special needs [9]. The sonification-based game Blindminton will be motivated from a brief analysis of sport games in general (Sec. 4.1), and games for people with visual impairment in particular (Sec. 4.2). We illustrate the progression of our started research project and then show the current status of the implementation (Sec. 4.3).

4.1   Perception and Action in Sport Games

People playing sport games have to deal with great demands on multi-modal perception for action control due to the extreme spatio-temporal constraints in the complex and dynamic environment of sport games. It is therefore a very important condition for top-performance to use effective strategies for information perception. As visual information is considered to be the most important information for action control in sport games, one of the major interests in cognitive research on anticipation and decision making in sport games is to analyze visual search strategies [10].

Despite the dominance of visual information there can be no doubt that other types of information are also important for top performance in sport games and you need a holistic, multi-modal perception for optimal action control. For

instance, you cannot reach top performance in table tennis without receiving auditory information about the ball bouncing on the table.

The human multi-modal perception system is adaptive to the environmental demands. On the one hand, this fact is used in training concepts closing one information channel (e.g. 'Close your eyes!') to train other perception systems ('try to control the ball just with the tactile information of your feet!'). On the other hand, people with visual impairment are forced to adapt their information processing due to their missing visual information in everyday life. This motivates us to search for sport games reducing the requirements of visual information and increasing the importance from other information like sound. Thus we take a look for already existing games for people with visual impairment.

4.2   Sport Games for People with Visual Impairment

Sport games offer important experiences in body movement and body motion and are of crucial importance for the psychosocial development of people with visual impairment. As visual information is the leading afferent information for action control in sport games it is particularly difficult for these people to take part in sport games. But since these people also desire to get access to sport games, it is one of the most important tasks in the research field of APA to expand the boundaries of ordinary sport games and search for new opportunities or enabling techniques to facilitate their participation.

Until now, there are only very few sport games for people with visual impairment. One of them is the so called ball game Goalball which is very significant: Goalball was created especially for blind people and the only paralympic sport game for people with visual impairment for many years[4]. It was accompanied by the game 'football-5-a-side' in the recent paralympic games in Athens 2004. These sport games show impressively the adapted perceptual skills of sportsmen using non-visual information and proof the possibility to play sport (and even ball) games without any visual information.

Searching for further non-visual sport games, we use insights from three areas: Firstly, existing sport games like goalball are analyzed and their basic principles like the sounding ball are used to create new games. Secondly, we regard successful applications of interactive sonification in auditory computer games. Games like 'Super Tennis' can be played against the computer just using auditory information and are very popular, in particular for people with visual impairment[5]. Finally, we take virtual simulations of sport games into consideration. Games like 'Virtual Table Tennis'[6] are games consisting of virtual simulation. They can be played against the computer or via internet against another opponent within VR using a virtual ball.

We break new ground in using the method of interactive sonification [4] to present auditory information as the leading information for action regulation in

---

[4] see http://www.ibsa.es/eng/deportes/goalball/presentacion.htm
[5] see http://www.audiogames.com
[6] see http://www.vtt.fi/multimedia/camball/camball.html

sport games. This enables us to present auditory information in a more systematic way as in the existing sport games using natural sounds like the ringing of a bell inside the goalball. Auditory computer games already use interactive sonification and can be seen as challenging games, predestined for motivating flow experiences[11]. But from the perspective of APA computer games do not offer movement experiences promoting motor development like sport games.

AcouMotion integrates interactive sonification, movement experiences and virtual game simulations and goes beyond hitherto existing systems. The system provides a technical basis offering new auditory sport games that can be played just by using non-visual sonification-based information with real motor activity. In the following we present the game concept of our first application *Blindminton.*

4.3   Blindminton - The Game Concept

The lack of adapted sport games for people with visual impairment makes it necessary to provide these people with ample information enabling them to conduct a challenging sport game. We show how to realize this by using AcouMotion. To attack the ultimate challenge of a multi-player game for blind people, we decompose the problem in smaller steps. This allows to treat smaller problems, and to develop a series of highly promising research platforms.

Basically, Blindminton is a game where a (here virtual) ball is being hit by a racket until it comes to a rest. If a player places the ball into 'out', or fails to hit back the ball properly, the opponent gets a point. The winner of the game is the player who first reaches 15 points.
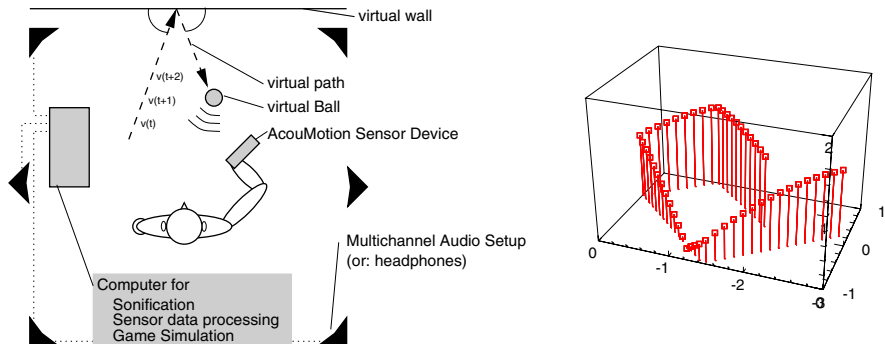
Fig. 2. Blindminton game setting. The plot shows a simulated ball trajectory with 4 impact events.

One-Player Blindminton. We start with a limited version of Blindminton where the opponent is replaced by a fixed wall, so that the task is being turned into the game of keeping the (virtual) ball in the game as long as possible. In this

game we can increase the score for every wall contact, and have the opportunity to make the score dependent upon the ball speed at the wall. This introduces an element which engages the player to increase his activity to obtain better scores. Alternatively, other motivational elements can be introduced, like the task to touch every tile of the virtual wall. What tiles remain to be touched can be communicated by the wall contact sonification which delivers the information at what distance and direction the nearest unhit tile is located. The score would then be reciprocal to the time needed to hit a tile.

All components of AcouMotion are required for implementing this game. In particular the AcouMotion sensor device is able to deliver position and orientation of the racket. Orientation is crucial since the ball reflects from the racket (via input angle=output angle) and this is an essential control to conduct the game.

Two-player Blindminton. Is an extension towards team plays. A second motion sensor device is needed. With some extensions, the rules of classical Badminton can be taken. The sonification engine has to be extended so that the opponents activity (e.g. position) is displayed to each player. Communicative aspects must be respected (like that the sonification may not overly interfere with vocal communication between players). This is the intermediate step towards team games like a '3 vs. 3' Blindminton (played like Volleyball).

4.4   Auditory Information Design

In games like Blindminton there are different types of information-carrying variables, like

Continuous Variables: Ball position, ball relative position, distance to racket, ball velocity, ball angular momentum, racket position, racket orientation,
Discrete Events: Ball/racket contact, ball/floor contact (in/out field), ball/wall contact, player (resp. sensor) leaves field borders,
Pseudo-discrete Events: Using a division of space into zones: ball crosses a zone plane. Pseudo-discrete events create an auditory gestalt, for instance as a pulsed event chain so that information is conveyed not only through the event itself but its relation to other events.

The auditory display aims at delivering much more information in sound than obtainable in real-world interactions. For instance, a flying ball may contribute a level-modulated sound pattern with pulse rate increasing when the ball approaches the racket and 'effet' can become audible as well. A video of our current Blindminton game is available on our website[7].

5   Discussion: Towards Acoustic Body-Tracking

The paper has introduced the AcouMotion system and Blindminton as a first application. AcouMotion opens a range of applications and research directions.

---

[7] see http://www.techfak.uni-bielefeld.de/∼thermann/projects/index.html

In **Sports Science**, AcouMotion plays a role to investigate for instance *motivational aspects*. Which informational factors (display richness, information latency, difficulty) make games motivating? Here the obtained information can be controlled via the sonification engine in full accuracy. Connected to this is the *analysis of motor laerning and training processes*: How do humans learn coordinated movements, and which factors can efficiently contribute to accelerate learning processes? For instance, it is argued that beginners learn faster if their task is simplified (e.g. by a larger racket). In Blindminton, we can not only control the virtual racket size dependent of the performance level, we can also control such variables continuously during the learning progress within the activity. Further on, AcouMotion has recently been used for testing players' reaction on spatially resolved sound cues. From a pilot study ([12]) we are optimistic to develop a useful performance test for paralympic game Goalball using AcouMotion. *Flow experience* is another important phenomenon observed in sports, but also in musical performance, etc [11]. It describes the dissolving of the person in his activity so that the mental focus becomes free to concentrate on higher levels (e.g. in playing music: from technical control to performance and emotional expression). The factors that potentially contribute to the emergence of flow can easily be examined with AcouMotion.

In the discipline of **Data Mining**, the challenging task is to understand structures in high-dimensional data. Interactive techniques can support the insight into data. Exploration and navigation tasks heavily rely on the perception-driven refinement of activity. Thus, AcouMotion may be applied to decrease the gap between abstract high-dimensional data spaces and human's natural interfaces, e.g. by using multi-modal exploration models that involve sonification. An impact on the degree of immersion, performance, reaction time, or a reduce of fatigue may be positive outcomes of applying AcouMotion in this domain.

In **Physiotherapy** we see the potential that AcouMotion can be a useful tool to induce therapeutically valuable movement patterns. To give only one example, consider a game where the tangible sensor device is used to catch virtual butterflies around your body. An additional sensor array attached to the user's back records motion parameters and directs the butterflies so that you activate your back in a therapeutically ergonomic way.

Cognitive Research. We believe that AcouMotion offers an interesting *analogy to eye tracking*: while the measurement of eye movements in response to visual events as a major 'window' into cognitive processes has become a widely established methodology, the analogous measurement of *body movements* in response to *auditory information* has so far been much less exploited. AcouMotion can fill this gap by providing a sound basis for studying this complementary link between modalities, complementing the dyadic eye-mind hypothesis of eye tracking research[13] with a triadic ear-mind-bodymotion hypothesis, stating that body motion responses to specific sound patterns can reveal information about the focus of ongoing cognitive processes. In this way, AcouMotion helps to answer research questions like: How are acoustic information from interactions processed and used to refine motor activity? How are emotional cues processed? Sound is

an ideal carrier for emotional information, and emotion influences body gestures. How do these systems relate to each other?

In conclusion, we recommend AcouMotion as a new auspicious platform to enhance human-computer interaction and investigate the relation between human information processing and human action.

Acknowledgment

References

1. T. Hermann and A. Hunt, "An introduction to interactive sonification," *IEEE Multimedia*, April-June 2005, vol. 12 no. 2, 20–24, IEEE.
2. G. Kramer, Ed., *Auditory Display - Sonification, Audification, and Auditory Interfaces.* Addison-Wesley, 1994.
3. A. Hunt and T. Hermann, Eds., *IEEE Multimedia, Special Issue Interactive Sonification*, IEEE, 04 2005.
4. T. Hermann, *Sonification for Exploratory Data Analysis*, Ph.D. thesis, Bielefeld University, Bielefeld, 2002.
5. C. Scaletti, "Sound synthesis algorithms for auditory data representations," in *Auditory Display*, G. Kramer, Ed. 1994, Addison-Wesley.
6. T. Hermann, J. Krause, and H. Ritter, "Real-time control of sonification models with an audio-haptic interface," in *Proc. of the Int. Conf. on Auditory Display*, R. Nakatsu and H. Kawahara, Eds. Int. Community for Auditory Display, 2002, pp. 82–86, Int. Community for Auditory Display.
7. J. McCartney, "Supercollider: a new real time synthesis language," in *Proc. ICMC '96*. Int. Comp. Music Assoc., 1996, http://www.audiosynth.com/icmc96paper.html.
8. H. Ritter, "The graphical simulation toolkit Neo/NST," `http://www.techfak.uni-bielefeld.de/ags/ni/projects/simulation_and_visual/neo/neo_e.html`, 2000.
9. G. Reid and H. Stanish, "Professional and disciplinary status of adapted physical activity," *Adapted Physical Activity Quarterly*, vol. 20, pp. 213–229, 2003.
10. A.M. Williams, K. Davids, J.G. Williams, *Visual Perception and Action in Sport*, E & F. N. Spon., London, 1999.
11. M. Csikszentmihalyi, *Beyond boredem and anxiety*, Jossey-Bass, San Francisco, 1975.
12. O. Höner, T. Hermann, T. Prokein, *Entwicklung eines goalballspezifischen Leistungstests*, In S. Würth et al. (Eds.), Sport in Europa, p. 331, Hamburg: Feldhaus, 2005.
13. M.A. Just, P.A. Carpenter, *The psychology of reading and language comprehension*, Boston: Allyn & Bacon, 1987

Constrained Gesture Interaction
in 3D Geometric Constructions

Arnaud Fabre, Ludovic Sternberger,
Pascal Schreck, and Dominique Bechmann

LSIIT, UMR 7005 CNRS-ULP,
Pôle API, Bd Sébastien Brant,
BP 10413, 67412 Illkirch,
Université Louis Pasteur, France
{fabre, sternberger, schreck, bechmann}@dpt-info.u-strasbg.fr

Abstract. This article aims to present a *constrained gestural interface*
which allows to easily experiment spatial geometry for educational pur-
poses. It consists in a bi-manual gestural language specifically designed
in order to simplify user's interaction in geometric constructions. As
the inherent complexity of geometry in 3 dimensions is combined with
the cognitive difficulty of interacting with virtual environments, we pro-
pose to constrain the interaction: hand postures constrain the object
type for designation and selection; objects already selected constrain the
construction process; the degrees of freedom representation constrain
the manipulation of constructed figures; and the deformable ray-casting
method constrain the navigation.

1   Introduction

When a geometric statement is given, it is not easy to find a construction satisfy-
ing the imposed properties, especially in three dimensions. A lot of 2D geometric
software products [1, 2, 3, 4, 5] have been designed for the CAE (Computer-Aided
Education) domain. They all define a geometry framework in which classical
tools, like ruler and compass, are translated into a computer representation, im-
proving capabilities of the pair pen/paper. Indeed, since a user is able to draw
some points and lines on a computer screen, it is just a stone's throw from mod-
ifying the figure by direct manipulation with keeping its properties verified. It
has been done by Cabri-géomètre [6, 7], one of the most famous 2D geometric
constructions application, which has defined the term of *Dynamic Geometry*.

However, the third dimension has been poorly tackled. Some software like
Cabri 3D [8], Geospacw [9], or Calques 3D [10, 11] can be picked out. But, their
repercussions into 3D geometry teaching have not been conclusive. In practice,
3D geometry visualisation and manipulation with 2D tools are not enough intu-
itive and efficient. Since the main problem of traditional space geometry appli-
cations is the gap between an object concept and its representation, the use of a
virtual reality environment enhances 3D geometric constructions comprehension
as the user is able to intuitively reach the implied geometric entities and interact
with them.

In an earlier paper [12], we have described our 3D geometric constructions prototype in a virtual environment, called Coyote-géomètre, which is a veiled reference to the famous Cabri-géomètre [6]. Our approach is based on the use of gloves to recognize gestures and postures. It is the main difference with previous studies like Construct3D by Hannes Kaufmann [13]. Applied in another domain, our approach is closer to the one of Zeleznik et al. [14] who proposed SKETCH, a gesture-based interface to "approximate" 3D polyhedral modelling.

As said above, the main goal pursued with the conception of Coyote consists in providing a better perception and comprehension of space geometry. But, our experience have proved that it remains inefficient to let too much abilities to the end user since the inherent complexity of geometry in 3 dimensions requires him to be guided in the construction process. Gesture interaction is a very intuitive and natural approach, although its many degrees of freedom (DOFs) are not easy to be apprehended. These two statements imply to find out solutions to constrain the interaction. We think that the use of a constrained gesture interaction would just decrease the complexity induced by the combination of VR and 3D geometrical constructions.

In section 2, an example of construction is presented. The section 3 shows how to create and select objects in a geometric scene by gestural interaction. In section 4, bi-manual construction is exposed. In section 5, the constrained manipulation of the resulting figure is studied. Finally, in section 6, the constrained navigation using the deformable ray-casting method is described.

2   Example of Construction

For example, let us consider the geometric universe composed by 6 types of objects : points, segments, lines, planes, circles, and spheres, and by 30 construction primitives, like construct a line passing by two points, etc. Although there is only few types of objects, it is sufficient for a relatively large choice of primitives and for an interesting use in CAE. Let us consider the following statement :

The intersection between a cube and a plane is at more a hexagon,

as there is no predefined object "cube" in our universe, we have to construct a cube.

A corresponding textual plan of construction of a wire cube could be the following (there is generally more than one plan of construction corresponding to the solution of a problem) :

1. Let p_1 be a point
2. Let d_1 be a line passing through p_1
3. Let d_2 be a perpendicular line to d_1 passing through p_1
4. Construct π as the plan passing through d_1 and d_2
5. Let p_2 be a point of d_1
6. Construct σ the sphere with center $p1$ passing through p_2
7. Let p_3 be one of the intersections of d_2 and σ
8. Let d_3 be a perpendicular line to the plane π passing through p_1
9. Let p_4 be one of the intersection of d_3 and σ
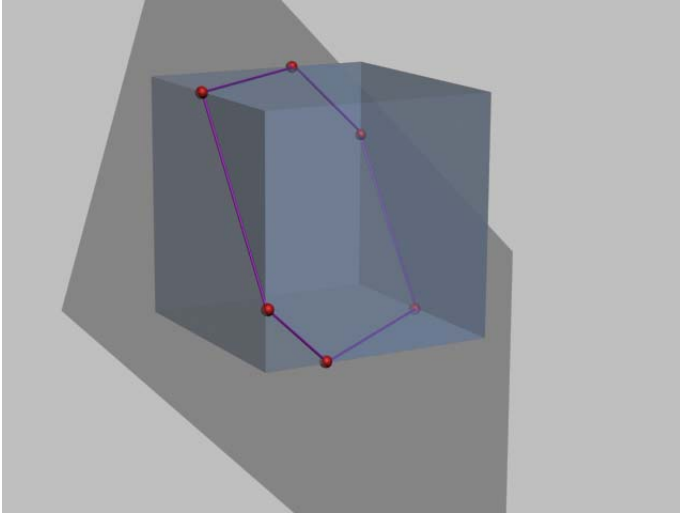
Fig. 1. Intersection of a cube and a plane

10. Construct the parallel line d_4 to d_1 passing through p_3
11. Construct the parallel line d_5 to d_2 passing through p_2
12. Let p_5 the intersection of d_4 and d_5
13. Construct the parallel line d_6 to d_1 passing through p_4
14. Construct the parallel line d_7 to d_3 passing through p_2
15. Let p_6 the intersection of d_6 and d_7
16. Construct the parallel line d_8 to d_4 passing through p_3
17. Construct the parallel line d_9 to d_2 passing through p_2
18. Let p_7 the intersection of d_8 and d_9
19. Let d_{10} be a perpendicular line to d_6 passing through p_6
20. Let d_{11} be a perpendicular line to d_4 passing through p_5
21. Let p_8 be the intersection of d_{10} and d_{11}

Then, a free plane is created and the intersections between the plane and the cube are computed while manipulating the plane. This example shows that some objects are free like $p1$, others are semi-defined like p_2, and other ones are totally defined like $p5$. So selection, creation and manipulation are different for each kind of objects. It also shows that a geometrical construction could be decomposed in two steps: the choice of the construction to be carried out, and the selection of implied objects. Finally, a pedagogical navigation is necessary to really see how to obtain a hexagon like in fig. 1.

This example will be used as a reference in all this article.

3   Bi-manual Creation and Selection

Achieving dynamic 3D constructions is not very simple. A stereoscopic display system, named Workbench could help the user in this task. A set of infrared

(a) Point gesture (b) Segment Gesture (c) Plane Gesture

(d) Circle gesture (e) Line gesture (f) Sphere gesture
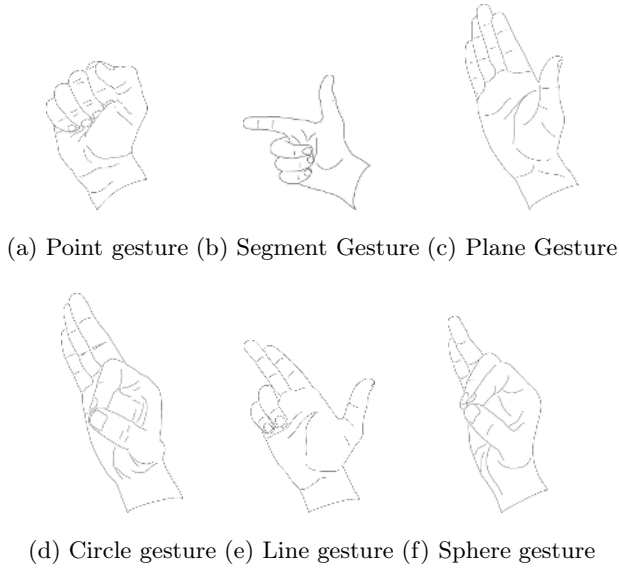
Fig. 2. Visual example of a Gesture Dictionary

cameras is used to follow his both head and hands movements by tracking positions and orientations. By wearing a pair of data gloves with the infrared trackers fixed on the back side of the wrist, user is able to do gestures and perform a construction.

Starting a new construction, the Workbench environment is empty. The user can freely add geometric objects, this is the creation step. Then, he can designate them for manipulation : the selection step and the manipulation one. Switching between creation, selection and manipulation modes is automatic. On the other hand, before navigating, the user has to explicitly change the mode.

The creation mode starts by pointing an empty place in the scene with the dominant hand, and closing the non-dominant hand at the same time. A new object is added and displayed in the virtual environment, depending on the user's dominant hand shape (fig. 2).

3.1   Creation

There are too many different objects in geometric constructions and their combinations are too numerous to have a single metaphor to interact with. When several techniques are used in the same time, each should visually correspond to a precise object, and act as an icon in mind. A meaning to interaction is provided by creating a vocable with our gestures set. Each posture has a visual signification : for example, a open hand is used to add a plane in the scene (fig. 3(b)). It is a so-called shape-based hand gestures language: the gesture's semantic is bounded to the commands' semantic and it acts like a semantic shortcut. To make the association, a *gesture dictionary* is defined (fig. 2). This is a list of
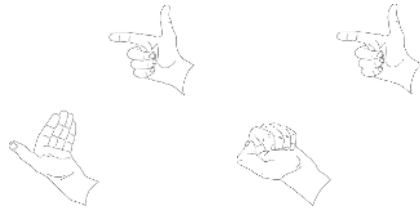
(a) Point designation (b) Point selection

Fig. 3. Parallel selection

gestures which represent the vocabulary for the application and its associated object.

Due to a small number of objects, we chose to create a dictionary with one gesture by geometric object. Each has been chosen to be far enough from the others, some have been given up because they were too ambiguous. Using Data-Glove from 5DT, measurements of joint angles and the spatial orientation of the hand can be determined. We used those values to recognise gestures in combination with 2 neural networks, one by hand. The first phase is to initialize the system and familiarize the subject with the system. Then, networks are trained by collecting data from user's gestures, which provide a profile by subject. The system records one profile by user and all profiles are used to make a default network for a new subject.

The fist gesture is used for adding/selecting a point (fig. 3(a)). The gesture associated with a segment is an index finger pointed out (fig. 3(b)). The line is represented by the index and the middle finger pointed out (fig. 3(e)). The hand open correspond to a plane (fig. 3(c)). The crab claw (the thumb and the index make a circle) represents of course the circle (fig. 3(d)). And if the middle finger is in our crab claw, it is the sphere (fig. 3(f)). Since there are few objects to be considered, the choice to make this dictionary is correct. But, if the user wants to add other objects - like a conic or a face - or macros-objects - like a cube composed of points, segments and faces -, a default designation representation have to be considered. The point gesture with the thumb down is proposed to be this representation.

3.2   Selection

Generally, selection is decomposed into two phases. The designation is the first phase, during which user points an object, non-dominant palm open. Then, in the second phase, he validates the selection by closing that hand (fist gesture). The application uses layers both to decompose the construction process, by hiding intermediate parts of construction for a better visibility of the solution, and group geometric objects by type to provide a new designation mechanism. On the one hand, progressive visualisation of the layers allows a step by step explanation of the construction to students. By providing a layer by type of object, the selection becomes non-ambiguous and parasitic objects are eliminated from
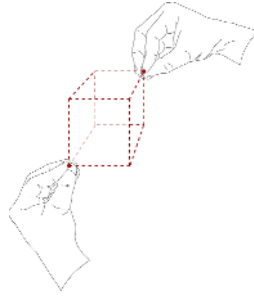
Fig. 4. Bounding Box

the picking for the selection. In this scheme, there is a correspondence between a gesture, a type of object and the current layer. On the other hand, multiple selection is possible directly selecting a layer without selecting objects. Finally, layers are gathered in a root layer, which provides an equivalent to the control-a hot key in a WIMP environment.

Multiple selection could use a bi-manual interaction. A volume can be selected by designing a bounding box (fig. 4). Every object which falls inside the bounding box becomes selected.

Designation by our gesture language have been described in this section. In next sections, the answer to the question "how to use this language to construct, manipulate and navigate ?" is provided.

4   Bi-manual Construction

In our prototype, the two steps of a construction defined in section 1 (selecting objects, then selecting a primitive of construction), can be done in any order: if a construction is chosen, the system asks for awaited objects, and if objects are selected, available constructions are restricted to those objects.

We noticed that those tasks are often put into parallel. Thus, a parallel use of hands to manipulate in the same time the geometric scene and the menu of constructions is proposed. While the dominant hand selects objects in the scene, the non dominant interacts with a contextual dynamic menu (fig. 5). It allows the user to work efficiently without considering all the possibilities of construction but only those he can effectively make. It consists in a semi-transparent menu and entries are selected via the non-dominant hand. The menu appears after a short delay (less than one second), allowing blind selection. For example, the user selects two points in space with his dominant hand. After one second, the menu appears and he selects an item (e.g. build the line passing by these points), by pushing his non-dominant hand through the menu. Moreover, it reduces the visual clutter of old large menus, especially when they are unrolled in hierarchical versions.

The non-dominant hand acts in the same time as a trigger in the menu, and as a trigger in space (geometric object selection).
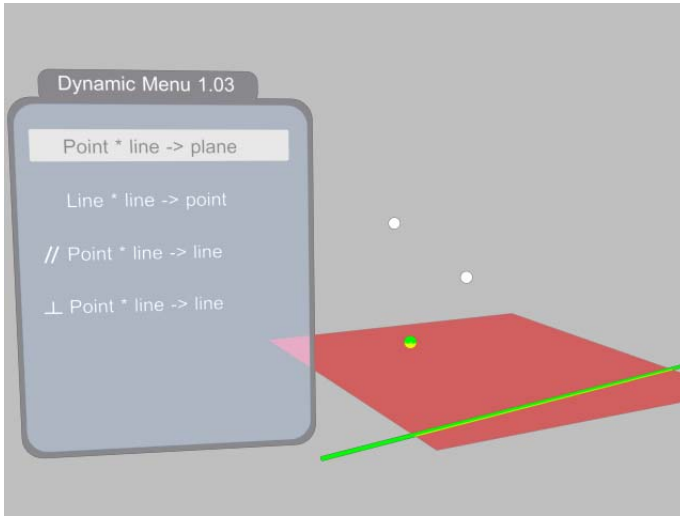
Fig. 5. Dynamic menu at the beginning of the construction

Imagine that the user selects a point and a line during a construction. The system can't know what the user want to do: build a parallel or a perpendicular. The construction is disambiguated by choosing in the dynamic menu the parallel icon with the non-dominant hand.

Remark: The traditional method of construction is to ask the system a possible list of construction primitives when a precise object is required. It can be done through a default menu.

5   Bi-manual Manipulation

With six degrees of freedom (3D position and orientation) manipulations are often very difficult, even if there are visual cues.

Bi-manual manipulation is a variant of direct manipulation that involves using both hands for a single task. According to Guiard's kinematic chain theory [15], the non-dominant hand provides a reference to user for improving his geometric construction.

The manipulation process can act on two kinds of objects. First, those directly created by the user (e.g. a point created freely in space). Second, those resulting from the construction process (e.g. a line created from two points). In the two next subsections, we propose two ways to help the user during manipulation and positioning.

5.1   Regular Magnetic Grid

The regular magnetic grid, illustrated in fig. 6, allows to place points, lines or planes with accuracy. Actually, each geometric object is *snap-dragged* on a node
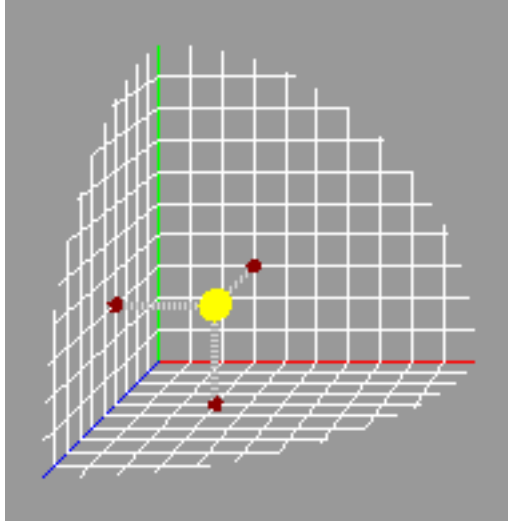
Fig. 6. Regular Magnetic Grid

of the mesh. Space between nodes can be adjusted via the non-dominant hand, making the grid more or less accurate (that hand has to catch one axis and bring it closer, or move away, from the center of the grid). With our Regular Magnetic Grid, problems of precision and depth have drastically decreased.

In our example (the intersection between a cube and a plane), we proposed to place a free point at first. This has to be done using the Regular Magnetic Grid. The plane could also be oriented by isothetic positioning.

5.2   Local Augmented Frame

We propose to add visual references to manipulate virtual objects. Several visual references are located on a Local Augmented Frame, and are called handlers (for scaling, rotating and translating) as depicted on picture 7.

That frame is materialized when an object is manipulated, by being centred on it. Different handlers corresponding to each kind of possible modification are represented in this visual cue. We distinguish displacements (rotations and translations) and scales. Cones are used to represent translational DOFs. Spheres are associated to rotational DOFs. Scaling by an axis is represented by a cube on it. Three small triangles are used to make translations in a plane.

In our example we would like to experiment the intersection between a plane and a cube. That plane could be moved in one direction (the normal direction) and has 2 RDOFs. So, moving a small cube provided with the Local Augmented Frame allows its translational movements, and rotations via the small spheres.
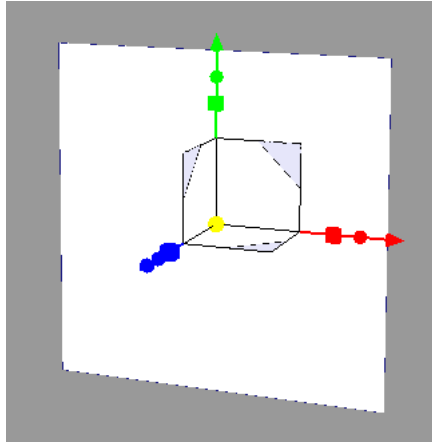
Fig. 7. Plane with local augmented frame

6   Constrained Navigation

Navigation in a geometrical scene is not very intuitive. Navigation is strongly correlated with physical displacements, but those have to be proscribed in the environments we are working for, since we have only a very limited space for walking. Moreover, basic components are not easily distinguishable, and the representation of their names considerably obstructs visual space.

The navigation is based both on hands movements and on the deformable ray-cast metaphor [16]. Our deformable ray-casting interaction technique allows traveling inside a virtual environment. The user simply draws a 3D path with the free form ray, and follows it by being attached to the curve. It provides a mean for planned navigation. We can see on fig. 8 a two-hands manipulation of the deformable ray. There are three different modes to use the ray: staying in place, moving forward (growing) and moving backward (shrinking). First, there is a dead zone when both hands are closer than a predefined distance (e.g. 0.1 meters). Second, the normal mode is enabled when both hands are further than the previous distance, and the non-dominant hand is opened. Third, when this last is closed, the ray shrinks in length. We need to precise that the deformable ray can't twist: it is just curved in space.

The dominant hand controls a 6 DOFs camera without zoom (i.e. it provides to the user a second point of view) which is fixed to the end of the ray. User

Fig. 8. Two hands manipulation of the deformable ray

is able to freely direct it in any direction by turning his wrist. By default, the camera is pointing forward of the fingers and attached to the 3D curve. The user can only change the rotation of the camera, and its instantaneous velocity on the curve is provided with the same technic than during the creation of the 3D path. Of course, it's not possible to rotate more than 180 due to the physical limitation of the front arm. On your previous example, a teacher could record a precise path to easily show how many segments (i.e. what kind of polyhedron) are generated by a plane/cube intersection. After, he will be able to play it again to his students during a learning session.

7   Conclusion and Future Works

A constrained gesture interaction has been presented for the domain of 3D geometric constructions. Its four main functionalities are :

- the possibility to constrain designation and selection to one type of object;
- a construction process constrained by previous selections;
- the constriction of manipulation by representing degrees of liberty by handlers;
- the use of the deformable ray-cast metaphor to constrain the exploration of a figure.

Currently, we are working in a relatively large environment : an Holobench composed by two screens, but, we project to port our application to standard PC equipment with low cost data gloves that could be used in classrooms. Learning space geometry in a intuitive way is the first step toward geometrical modeling in CAD (Computer-Aided Design) and in particular in the sketching of objects.

As accurate values are also needed to construct useful objects, the 3D constraints definition in the GCSP (Geometric Constraint Solving Problems), could be easier with such an interaction and it is one of the principal motivation with regard to the continuation of our research.

References

1. Roland Mechling. Euklid. http://www.mechling.de, a deutsch geometric constructions software (Shareware), 1994.
2. Nicholas Jackiw. The geometer's sketchpad. Key Curriculum, Berkeley, 1991-1995.
3. Ulrich Kortenkamp. Foundations of Dynamic Geometry. PhD thesis, Swiss Federal Institue of Technology, Zurich, 1999.
4. Stéphane Channac. Conception et mise en oeuvre d'un système déclaratif de géométrie dynamique. PhD thesis, Université Joseph Fourier - Grenoble 1, 1999.
5. L. Trilling, R. Allen, and S. Channac. The role of requirements, specifications, and implementation in constructing dynamic figures. Journal of Computers in Mathematics and Science Teaching 19(3), 195-209, 2000.
6. Yves Baulac. Cabri-géométre, a tool for computer aided geometry. Wheels for the mind. 5 (1), 30-34., 1991.

7. Laborde Colette, Laborde Jean-Marie. Problem solving in geometry: from microworlds to intelligent computer environments. Mathematical Problem Solving and New Information Technology (pp.177-192). Springer Verlag (1992).
8. Safwan Qasem. Conception et Réalisation d'une Interface 3D Pour Cabri-Géomètre. PhD thesis, Université Joseph Fourier - Grenoble 1, 12 décembre 1997.
9. Geospacw. http://www2.cnam.fr/creem/, 1998.
10. N. Van Labeke. Calques 3d: a microworld for spatial geometry learning. ITS'98 - System Demonstrations, San Antonio (Texas), August 16-19, 1998.
11. Nicolas Van Labeke. Prise en compte de l'usager enseignant dans la conception des EAIO. PhD thesis, Université Henri Poincaré Nancy, 1999.
12. Arnaud Fabre, Pascal Mathis, Pascal Schreck. 3D Geometric Constructions in Virtual Reality . IEEE Virtual Reality International Conference, Laval Virtual, May 2004.
13. Hannes Kaufmann. Construct3D: An Augmented Reality Application for Mathematics and Geometry Education. In Proceedings of ACM Multimedia Conference 2002. 2002.
14. Robert Zeleznik and K.P. Herndon and J.F. Hughes. SKETCH: An Interface for Sketching 3D Scenes. ACM Transactions on Graphics, Proceedings of SIGGRAPH'96. 1996.
15. Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. Journal of Motor Behavior, 19, 486-517 (1987).
16. Ludovic Sternberger, Dominique Bechmann. Deformable Ray-Casting Interaction Technique. In Proceedings of YoungVR 2004, 2005.

Gestural Interactions for Multi-parameter Audio Control and Audification

Thomas Hermann[1], Stella Paschalidou[2], Dirk Beckmann[1], and Helge Ritter[1]

[1] Neuroinformatics Group,
Faculty of Technology, Bielefeld University, D-33501 Bielefeld, Germany
[2] TEI of Crete - Rethimno, Greece
Department of Music Technology & Acoustics,
E. Daskalaki - Perivolia, 74100 Rethimno, Greece

Abstract. This paper presents an interactive multi-modal system for real-time multi-parametric gestural control of audio processing applications. We claim that this can ease the use / control of different tasks and for this we present the following as a demonstration: (1) A musical application, i.e. the multi-parametric control of digital audio effects, and (2) a scientific application, i.e. the interactive navigation of audifications. In the first application we discuss the use of PCA-based control axes and clustering to obtain dimensionality reduced control variables. In the second application we show how the tightly closed human-computer loop actively supports the detection and discovery of features in data under analysis.

1 Introduction

In the domain of audio processing systems, the usually high dimensional data space of the control task cannot be easily managed in real-time with the traditional human-computer interfaces used for office work tasks, e.g. the drop-down-menu-selection of the mouse-paradigm. A suitable gestural approach can make the control task more effective and intuitive, and expand the system's capabilities. The new gestural system we present is described in detail in Section 2. A general problem in using gestures for interactive control tasks lies in 'mapping' [1] the vector space of the gestural parameters to the vector space of the control parameters. 'Explicit mapping' [1] usually relies on the developer's intuition, which doesn't always lead to the most effective solution for the control task. In Section 3 a more intelligent and adaptive approach of this procedure is under exploration; Principle Component Analysis (PCA) of the control data space distribution is applied to obtain dimensionality reduction. Section 4 introduces the application of our system to data *audifications* [3]. We demonstrate enhanced gestural interaction with synthetic and seismic data. The paper closes with a discussion of our experiences and an outlook of future work.

2 System Overview

Our Gestural interaction system provides a closed-loop human computer interface. A webcam/EyesWeb subsystem described in [2] is used as sensor. EyesWeb
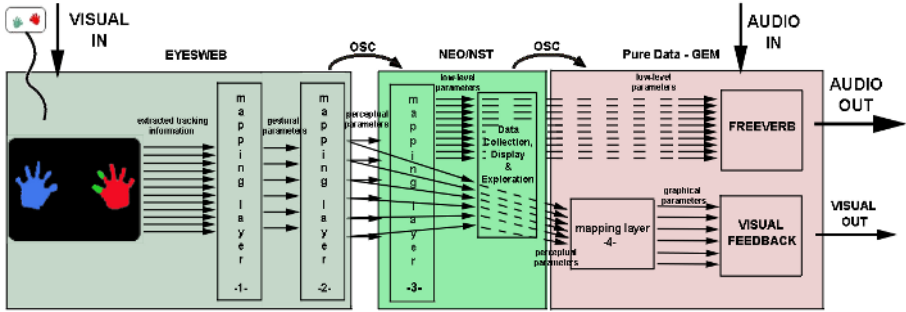
Fig. 1. Scheme of the System Architecture. Modules communicating via OSC.

delivers the two hand coordinates and bounding boxes for recognized hand blobs currently distinguished by using colored gloves. Gesture definition can be done either in EyesWeb, or alternatively in Pd[1] or Neo[2]. Open Sound Control (OSC) is used to connect these modules.

The gestural controls are mapped to high-level controls (like spaciousness or warmth) which are mapped to low-level controls (or output features) required by the application (e.g. reverb parameters in DAFx control, effect filter frequencies in audification) via a manually tuned control function and sent to the application module via OSC. The modular system architecture allows to distribute the system on different computers, which might be necessary if the modules are computationally demanding.

3   Controlling Audio Effects - LPCA-Based Mapping

In the past an attempt was made by the second author to develop a functional multi-modal real-time environment for "Alternative Control of Digital Audio Effects" [2], in specific a reverb[3] as a prototype. The challenge was to define suited hand gestures, to control the 10-dimensional output feature space of the reverb in a more intuitive, engaging and creative way. Nevertheless, all mapping layers had to be defined based totally on intuition and trials. Often few uncorrelated dimensions suffice to approximate an output feature at sufficient accuracy. From this observation, the question arises whether the vector space of useful controls can be segmented into regions that form natural anchors for the control task. This section suggests local principal component analysis (LPCA) as a means to obtain such a segmentation on a data-driven way.

In the case of the reverb prototype, we gathered data from typical control situations and then used these to compute the principal axes. Fig. 2 shows a plot

---

[1] PureData by Miller Puckette.

[2] Neo/NST, graphical programming environment developed in Bielefeld, see [4].

[3] 'Freeverb', created by Jezar Wakefield at Dreampoint Design and Engineering.
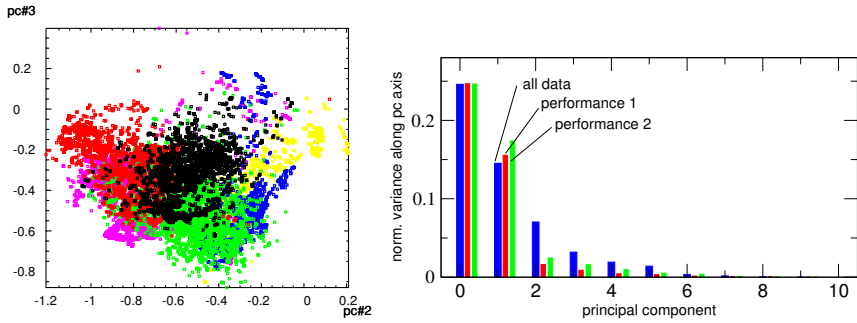
Fig. 2. PCA plot of low-level control vectors for different performances. The right plot shows eigenvalue spectra for all controls, and for two selected performances.

of the low-level parameters in the reverb application (projected on some principal axes). It can be seen that the different performances cluster. The effective control dimensionality may be seen from the eigenvalue spectrum plotted right – in isolated control presets, even 2-3 control dimensions would suffice to cover all points accurately. We propose as gesture mapping, to connect gestures that are easy to perform and independent, with variation along the principal axes of highest variance. Mathematically this is achieved by computing the output feature vector $\mathbf{y}$ by $\mathbf{y} = A\mathbf{x} + \mathbf{b}$, where $\mathbf{b}$ is the prototype center, $\mathbf{x}$ the gestural controls, A the matrix of eigenvectors (as columns) to the largest eigenvalues of the data covariance matrix computed for those data used to determine the preset.

4   Gesture Controlled Audification

Our scientific application aims at interactive search of 'interesting' events in data sonifications. The human ear is highly trained to perceive patterns even in very noisy signals, however, this capability is limited to well known masking thresholds. Spectral filtering is therefore a practical technique to isolate parts of the audification and make specific elements more salient. But typically, parameter settings for available variables like the sampling rate, the playback speed and eventually spectral filtering parameters are set in advance. Thus, rendering of sonification and playback are separated steps.

We argue that an interactive audification, where the user is embedded in a closed control loop, may support pattern detection and accelerate the process of finding appropriate control settings. It demonstrates that the expressiveness of gestures is ideal for interacting with the multitude of parameters in interactive audification, and thus increase the usability of audification significantly.

In this application EyesWeb is used for recording and processing the incoming webcam images. The derived information, like hand positions (x, y) and palm sizes (w, h) for both hands, are sent to Pd. This alone offers at least 6D real-time control. We demonstrate the interactive audification system with two interaction videos where the gestural interaction and the sonification can be observed during

exploration of seismic measurements, and a synthetic data set where an initially hidden acoustic structure is discovered. Further information and the videos are available on the website [5].

5   Discussion

Gestural interactions provide rich means for real-time control, particularly in the case of audio systems. Instead of using gestures to directly control a low-level parameter, we have demonstrated two alternative ways of how to use intermediate structures: the first approach uses a definition of high-level parameters and a fixed connection to low-level parameters by means of a functional relation, while the second approach relies on a data-driven definition of control clusters and their prevailing control variable axes.

While gesture recognition systems and control applications grow in perfection, the intelligence between them attains higher and higher relevance. We think and hope that the application of data mining techniques in the domain of both the human gestural input and the algorithmic control parameter space can contribute to the identification of better suited gestures and the optimization of the mapping implementation, thus finally helping to the creation of more effective human computer interfaces.

References

1. R. Kirk A. Hunt, M. Wanderley, "Towards a model for instrumental mapping in expert musical interaction," in *Proceedings of the 2000 International Computer Music Conference*, San Fransisco, 2000, International Computer Music Association, pp. 209–212.
2. S. Pashalidou, "Alternative control of digital audio effects," M.S. thesis, University of York, York, UK, 2003.
3. F. Dombois, "Using audification in planetary seismology," in *Proc. of the 7th Int. Conf. on Auditory Display*, Nick Zacharov Jarmo Hiipakka and Tapio Takala, Eds., Helsinki University of Technology, 2001, ICAD, pp. 227–230, Laboratory of Acoustics and Audio Signal Processing.
4. H. Ritter, "The graphical simulation toolkit Neo/NST," `http://www.techfak.uni-bielefeld.de/ags/ni/projects/simulation_and_visual/neo/neo_e.html`, 2000.
5. T. Hermann, "Sonification for exploratory data analysis–demonstrations and sound examples," `http://www.techfak.uni-bielefeld.de/~thermann/projects/index.html`, 2002.

Rapid Evaluation of the Handwriting Performance for Gesture Based Text Input

Grigori Evreinov and Roope Raisamo

Department of Computer Sciences, FIN-33014 University of Tampere, Finland
{grse, rr}@cs.uta.fi

Abstract. Rapid method for evaluation of pen-based text input techniques is necessary both for designers and consumers. We present a method that is based on an immediate performance comparison of the gesture making using the graphic templates of typefaces and the pen-based behavioral patterns. The results showed that besides the cognitive difficulty of symbolic gestures, metaphors and mnemonics, first and foremost the graphic feasibility determines handwriting performance of the gesture-based input techniques.

1   Introduction

While pen-based text entry is becoming a mainstream technology in mobile computing appliances, designers still try to embed QWERTY keyboard into PDAs and Smartphones. A smaller number of physical keys or software buttons requires learning and skills. A combination of fingers, when chorded keyboard is employed, is primarily based on the user's cognitive capabilities to recall and activate a group of the buttons in a definite position and moment of time. Another tendency in pen-based text input is to display all characters and then optimize layout for shorthand or continuous writing [6, 8, 9]. For an individual user a statistically optimal gesture may be inconvenient. Small circular motions require an accuracy and strain as inhibition processes dominate over excitation in such a case. Even linear motion can be sweet in one direction for some person and another one will prefer another direction. A strong motivation is necessary that a consumer really wishes to learn a new alphabet or the system of gestures when starting to use a new device [5]. How to predict that training and time will be not wasted due to individual unavailability of the gesture-based technique?

Speed in pen-based text input depends on the individual handwriting skills and finger dexterity. Length of traces and gaps between characters (lifting) are also essential restricting factors of the speed. Cognitive components, like attention and memory, language comprehension and reasoning also affect learning of motor skills, can facilitate or inhibit the memorizing and using gestures for human-computer interaction. While many authors recognize these factors which have an impact on making gestures, the special method for testing different graphic templates of typefaces and the system of behavioral patterns still stays beyond designing.

2   New Gesture Template and Testing

Western script has several handwriting styles for elementary school-aged children. The loops and other forms provide systematic steps for letter analysis and efficient

motor and memory cues for children. These strategies simplify the learning of functional handwriting [3]. To employ prior handwriting experience of the user and to avoid some restrictions in gesture recognition, some time ago Graffiti text input system was proposed as a commercial product of the Palm Computing Division of US-Robotics. Graffiti has been examined in a number of previous studies [5]. These evaluations have considered many important factors, such as novice vs. expert performance, the speed-accuracy trade-off, text creation vs. text copy tasks, focus of attention, quantitative vs. qualitative measures. However, is it possible to predict usability of different gesture-based techniques in a single case study without long-term exploration?

Let us suppose we have created a new form of character input based on a stylus gestures which are comprised of movements preferably drawn in one or two directions, for instance, from bottom left to top right position or/and backwards (Figure 1).
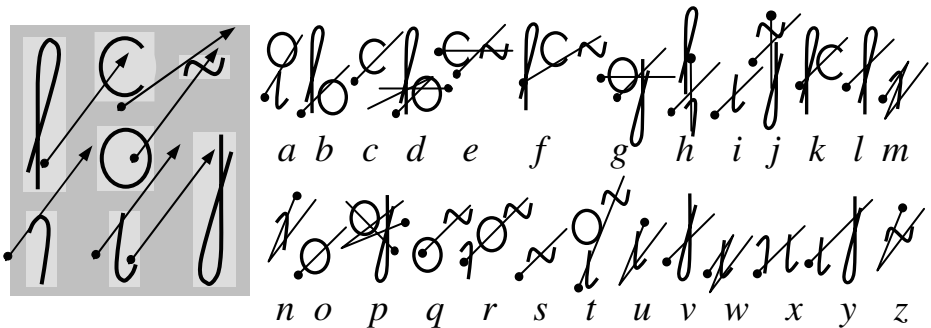
Fig. 1. The layout of segments for Symbol Creator and possible entry gestures (the arrows)

The black points in Figure 1 indicate the starting position for each gesture. Intermediate segments are ignored; some characters have alternative gestures. Most of the characters composed from selected (by crossing) segments are naturally connected like a conventional cursive typeface (Latin) in a low-case position. We named this technique as Symbol Creator and it was empirically evaluated earlier as the alternative on-screen keyboard for gaze-based text input [7]. The Symbol Creator software for pen-based text input was written in eMbedded Visual Basic 3.0 and preliminarily tested on an iPAQ pocket PC.

Individual capability to make different gestures has to correlate with the immediate performance of copying graphic patterns and could be used as the first-hand consumer guidance in acquiring one or another software product for text entry purposes.

In the pilot tests we found out that of about 500 ms was enough to copy any image of different typefaces. Thus, 10 seconds for a trial might stimulate the subject to perform the test as fast as possible without any 'additional thinking' of how to do it. During this time at least 20 of 30 images would be copied in the template. To compare different pen-based text input methods we have decided to simplify the procedure to record copying of graphic templates without recognition of the gestures. The "bad" traces were removed automatically if the number of track points in a gesture array was less than one half of the number of points in the sample and the time of

making gesture was less than 50 ms. In such a way we avoided the long procedures of calibration to support different gesture recognition techniques.

The testing program SpeedGraph was written in Microsoft Visual Basic 6.0. Data were gathered using Fujitsu Siemens LifeBook CE0122X with touchscreen and a stylus for interaction. Four graphic templates were tested. They were the following: gestures for Symbol Creator (SCreator), gestures for "minimal device-independent text input method" – an abbreviated title is MDITIM [4], Unistrokes [2] and Graffiti [1]. 20 unpaid volunteers took part in the test using the SpeedGraph software with the four graphic templates of the gestures intended for pen-based text input. Seven subjects had prior experience with a pen-based computer and could operate with Unistrokes and Graffiti. Five subjects were female, 15 were male. The mean age of the subjects was 30 years. The complete testing duration was not more than 10 minutes per person: 10 attempts $\times$ 10 s $\times$ 4 templates, instructions, breaks and data preview.

4 Results and Discussion

Most of the subjects commented that MDITIM gestures were very difficult for copying. Drawing of Graffiti also showed a low speed - 16 images for 10 seconds or of about 380 ms per image at the standard deviation of about 400 ms. MDITIM patterns required of about 470 ms per image at the standard deviation of about 490 ms. While copying of the SCreator and Unistroke gestures, the speed difference was not significant (F < 0.025). However, the standard deviation was lower during copying of images for SCreator gestures (Figure 2). On the other hand in Unistroke gestures, only 5 patterns are similar to the conventional letters and some learning for other traces would have been required. In Symbol Creator only simpler movements in one or two directions have been used (Figure 1). At least 20% of the presented patterns were unconventional for the subjects (a, v, k, q, z). Therefore, we suppose that the copying of gesture images was not reproduced from the indicated starting position without any thinking. By pointing the starting position in the graphic sample, we provoke the
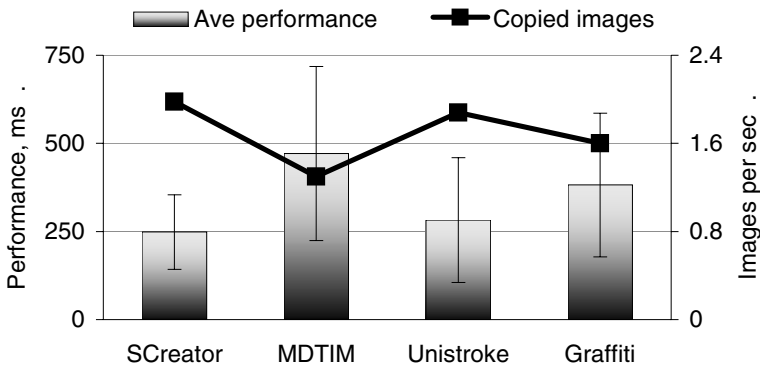

Fig. 2. The average performance throughout all tests (left scale), standard deviation and the number of copied images per second (right scale)

person to run a cognitive analysis before the gesture will be activated with a stylus. The direction of detour of the sample could also depend on individual preference.

It was equally probable that the number of loops could essentially increase the length of traces. In a case of mental pre-processing, we could observe increasing of inter-stroke gaps. However, analysis of graphic templates showed that the stylus was held on the touchscreen and was lifted for approximately equal time, when copying the images in four tested templates. As correlation between the total times of the drawing and the length of the images of the graphic templates was of about 0.73, the probability of mental pre-processing was lower than individual capability to make small circular movements with MDITIM and Graffiti gestures.

Thus, data analysis of four pen-based text input techniques showed that the type of motion and direction used for continuous writing are essential to produce any gestures intended for a higher performance interaction style, at least, with the use of a stylus and touchscreen. A person is expected to prefer more familiar and simpler gestures that require smaller cognitive efforts and fingers dexterity.

The method for rapid evaluation of new gestures for pen-based text input is based on a comparison of the personal immediate performance in copying the graphic patterns. The method does not require memorizing of the new gestures nor any specific layouts or rules before testing. Testing takes no more than two minutes per graphic template that is composed of 30 samples of the behavioral patterns.

References

1. Blickenstorfer, C. H.: Graffiti: Wow!!!! Pen Computing Magazine, 1 (1995) 30-31.
2. Goldberg, D., Richardson, C.: Touch-typing with a stylus. In: Proc. of the ACM Conference on Human Factors in Computing Systems - INTERCHI '93, New York, ACM (1993) 80-87.
3. Handwriting styles, fontware. Available at: http://www.educationalfontware.com/
4. Isokoski, P. and Raisamo, R.: Device Independent Text Input: A Rationale and an Example. In: Gesù, V.D., Levialdi, S. and Tarantino, L. (eds.): Proceedings of the Working Conference on Advanced Visual Interfaces AVI 2000, Palermo, Italy (2000) 76-83.
5. MacKenzie, I.S., Zhang, S.: The immediate usability of Graffiti. In: Proceedings of Graphics Interface '97. Toronto, Canadian Information Processing Society (1997) 129-137.
6. Mankoff, J., Abowd, G.D.: Cirrin: A word-level unistroke keyboard for pen input. In: Proceedings of UIST '98. Technical notes (1998) 213-214.
7. Miniotas, D., Spakov, O., Evreinov, G. Symbol Creator: An alternative Eye-based Text Entry Technique with Low Demand for Screen Space. In: Proceedings of INTERACT'03, Zurich, Switzerland, IOS Press, IFIP (2003) 137-143.
8. Quikwriting. Product information is available at: http://mrl.nyu.edu/projects/quikwriting/
9. Zhai, S., Kristensson, P-O., "Shorthand Writing on Stylus Keyboard". In: Proceedings of CHI 2003, ACM Conference on Human Factors in Computing Systems. (2003) 97-104.

Author Index